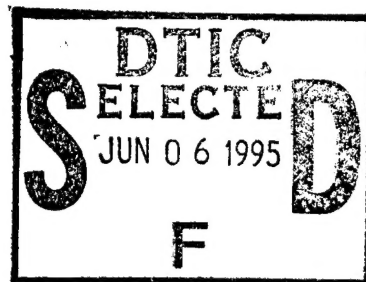


NAVAL POSTGRADUATE SCHOOL

Monterey, California



THESIS

**CLOSED FORM EXPRESSIONS FOR THE
PROBABILITY OF DETECTION FOR AN
ENVELOPE DETECTION APPROXIMATION
GO CFAR PROCESSOR**

by

Clement Tanaka

December 1994

Thesis Advisor:

Phillip E. Pace

Approved for public release; distribution is unlimited.

DTIC QUALITY INSPECTED 3

19950602 020

| REPORT DOCUMENTATION PAGE | | | Form Approved OMB No. 0704-0188 | |
|--|--|---|---|---|
| Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instruction, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188) Washington DC 20503. | | | | |
| 1. AGENCY USE ONLY (Leave blank) | | 2. REPORT DATE December 1994 | | 3. REPORT TYPE AND DATES COVERED Master's Thesis |
| 4. TITLE AND SUBTITLE CLOSED FORM EXPRESSIONS FOR THE PROBABILITY OF DETECTION FOR AN ENVELOPE DETECTION APPROXIMATION GO CFAR PROCESSOR (U) | | | 5. FUNDING NUMBERS | |
| 6. AUTHOR(S) Tanaka, Clement | | | | |
| 7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Naval Postgraduate School Monterey CA 93943-5000 | | | 8. PERFORMING ORGANIZATION REPORT NUMBER | |
| 9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) | | | 10. SPONSORING/MONITORING AGENCY REPORT NUMBER | |
| 11. SUPPLEMENTARY NOTES The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government. | | | | |
| 12a. DISTRIBUTION/AVAILABILITY STATEMENT Approved for public release; distribution unlimited. | | | 12b. DISTRIBUTION CODE | |
| 13. ABSTRACT (Maximum 200 words) The greatest of constant false alarm rate processor (GO CFAR) is a useful architecture for adaptively setting a radar detection threshold in the presence of clutter edges. The GO CFAR input is often the envelope detected in-phase (I) and quadrature (Q) channels of the baseband signal ($x_c = \sqrt{I^2 + Q^2}$). This envelope detection can also be approximated using $x = a I + b Q $ which requires less complex hardware (a and b are simple multiplying coefficients). The envelope GO CFAR processor and several envelope approximation GO CFAR processors are compared in terms of the probability of detection (PD) performance. Closed-form expressions which describe the PD performance are given and their accuracy evaluated. | | | | |
| 14. SUBJECT TERMS CFAR, CFAR Detector, CFAR Processor, Constant False Alarm Rate, Signal Detection | | | 15. NUMBER OF PAGES 71 | |
| | | | 16. PRICE CODE | |
| 17. SECURITY CLASSIFICATION OF REPORT Unclassified | 18. SECURITY CLASSIFICATION OF THIS PAGE Unclassified | 19. SECURITY CLASSIFICATION OF ABSTRACT Unclassified | 20. LIMITATION OF ABSTRACT UL | |

NSN 7540-01-280-5500

Standard Form 298 (Rev. 2-89)
Prescribed by ANSI Std. Z39-18

Approved for public release; distribution is unlimited.

**CLOSED FORM EXPRESSIONS FOR THE PROBABILITY OF DETECTION
FOR AN ENVELOPE DETECTION APPROXIMATION GO CFAR PROCESSOR**

Clement Tanaka
Lieutenant, United States Navy
B.S.M.E.N.E., University of California, Berkeley, 1984


Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN AERONAUTICAL ENGINEERING

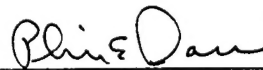
from the

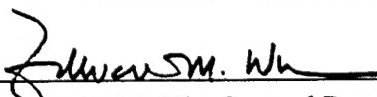
**NAVAL POSTGRADUATE SCHOOL
December 1994**

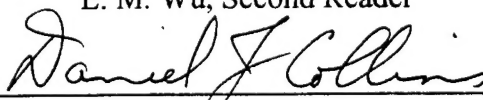
Author:


C. Tanaka

Approved by:


P. E. Pace, Thesis Advisor


E. M. Wu, Second Reader


D. J. Collins, Chairman
Department of Aeronautics and Astronautics

| | |
|--------------------|--|
| Accession For | |
| NTIS CRA&I | <input checked="checked" type="checkbox"/> |
| DTIC TAB | <input type="checkbox"/> |
| Unannounced | <input type="checkbox"/> |
| Justification | |
| By | |
| Distribution / | |
| Availability Codes | |
| Dist | Avail and/or Special |
| A-1 | |

ABSTRACT

The greatest of constant false alarm rate processor (GO CFAR) is a useful architecture for adaptively setting a radar detection threshold in the presence of clutter edges. The GO CFAR input is often the envelope detected in-phase (I) and quadrature (Q) channels of the baseband signal ($x_e = \sqrt{I^2 + Q^2}$). This envelope detection can also be approximated using $x = a|I| + b|Q|$ which requires less complex hardware (a and b are simple multiplying coefficients). The envelope GO CFAR processor and several envelope approximation GO CFAR processors are compared in terms of the probability of detection (PD) performance. Closed-form expressions which describe the PD performance are given and their accuracy evaluated.

ACKNOWLEDGEMENT

I would like to offer my sincere thanks to my advisor Professor Phillip Pace for providing the guidance, patience and encouragement to enable me to complete this project. I am also grateful to my second reader Professor Edward Wu who sparked my initial interest in probability theory and its applications to real world phenomena. Last but not least, I am eternally indebted to my wife Liz for her unending support, love and understanding throughout this endeavor.

TABLE OF CONTENTS

| | |
|---|----|
| I. INTRODUCTION | 1 |
| A. GO CFAR PROCESSOR | 1 |
| B. PRINCIPLE CONTRIBUTIONS | 1 |
| C. THESIS OUTLINE | 2 |
| II. CFAR PROCESSOR | 3 |
| A. BASIC SIGNAL DETECTION PRINCIPLES | 3 |
| B. GO CFAR PROCESSOR | 4 |
| C. NOISE AND PROBABILITY | 6 |
| 1. Probability Density Functions | 6 |
| 2. Cumulative Distribution Functions | 7 |
| 3. Summation of Density Functions | 8 |
| 4. Maximum of Two Density Function | 8 |
| 5. Linear transformation of density functions | 8 |
| 6. Absolute value of a density function | 9 |
| D. ENVELOPE DETECTOR AND APPROXIMATIONS | 9 |
| E. PROBABILITY OF FALSE ALARM VS THRESHOLD MULTIPLIER | 12 |
| F. PROBABILITY OF DETECTION VS. SNR | 14 |
| III. MONTE CARLO SIMULATIONS USING MATLAB | 17 |
| A. MATLAB PROGRAM | 17 |
| B. PROGRAMMING TECHNIQUES | 19 |
| IV. APPROXIMATION FORMULAS FOR PD | 25 |
| A. METHODS OF CURVE FITTING | 25 |
| B. ERF FUNCTION APPROXIMATION | 25 |
| C. TABLES OF COEFFICIENTS | 29 |
| V. PD APPROXIMATION COEFFICIENTS | 33 |
| VI. CONCLUSIONS | 41 |
| APPENDIX - MATLAB PROGRAMS | 43 |
| REFERENCES | 57 |
| INITIAL DISTRIBUTION LIST | 59 |

I. INTRODUCTION

A. GO CFAR PROCESSOR

The greatest of constant false alarm rate (GO CFAR) processor is a useful architecture for adaptively setting a radar detection threshold in the presence of clutter edges. The inputs to the processor often involve the envelope detected in-phase (I) and quadrature (Q) channels of the baseband signal $x_e = \sqrt{I^2 + Q^2}$. This envelope detection can be approximated using $x = a \max\{|I|, |Q|\} + b \min\{|I|, |Q|\}$ which requires less complex hardware and where a and b are simple multiplying coefficients (Pace, 1994 and Hache, 1994). This approximation can be simplified further to $x = a|I| + b|Q|$ which reduces hardware requirements even more. That is, the two channels do not have to be compared to determine the minimum and the maximum. Ironically, this reduction in complexity has increased the difficulty of obtaining closed form analytical solutions to the GO CFAR performance in terms of probability of false alarm (PFA) and probability of detection (PD). Recent studies have used numerical and Monte Carlo methods to detail the performance of this processor in terms of the PFA as a function of the threshold multiplier. Curve fit solutions have also provided closed form expressions for the PFA (Pace, 1994). Closed form expressions for the detection performance however, have not yet been examined.

B. PRINCIPLE CONTRIBUTIONS

This paper extends the results from previous studies and develops closed form expressions for the PD performance as a function of the signal-to-noise ratio (SNR). These results provide system designers with an accurate estimate of system performance for several detector approximations without requiring numerical analysis or Monte Carlo simulations. Since no closed form expressions for the PD exist, Monte Carlo simulations are first used to obtain a full set of the PD curves using a PFA of 10^{-4} for the number of reference cells $n=1,2,4,8,16,32$ and for various values of multiplying coefficients a and b .

Various curve-fit techniques are then used to obtain closed form expressions for the PD vs. SNR curves. Several formulations are investigated. The most promising expression involves the use of the *erf* function which produces a detection performance within ± 0.015 of the actual PD.

Next, the coefficients for each envelope approximation PD performance are plotted as a function of the number of cells n . This allows the extraction of the appropriate coefficients for any number of reference cells n between 0 and 32. This in turn provides a quick and relatively accurate expression for the PD as a function of the SNR for the various detection approximations a and b .

C. THESIS OUTLINE

Chapter II starts with a review of basic signal detection techniques, an introduction to the GO CFAR processor, and the concepts of noise and probability. It continues with a description of envelope detection as well as several approximations to the envelope detector and concludes with an analysis of the GO CFAR processor performance in terms of the PFA and the PD. Chapter III provides a discussion of the Monte Carlo techniques and the simulation results. Chapters IV presents the derivation of closed form expressions for the detection performance and provides tables of the corresponding coefficients. Chapter V presents the plots of the closed form expression coefficients as a function of the number of cells n . The appendix contains the code developed for this research.

II. CFAR PROCESSOR

A. BASIC SIGNAL DETECTION PRINCIPLES

The detection process for a typical radar system involves an inherently straightforward comparison between the received signal and a reference value. In a simple detector, the reference value is a *fixed threshold* which remains unchanged through the detection process. If the received signal is above the threshold, a target is assumed to be present, otherwise the signal is ignored. Not surprisingly, this is referred to as *threshold detection*. Figure 1 illustrates this concept.

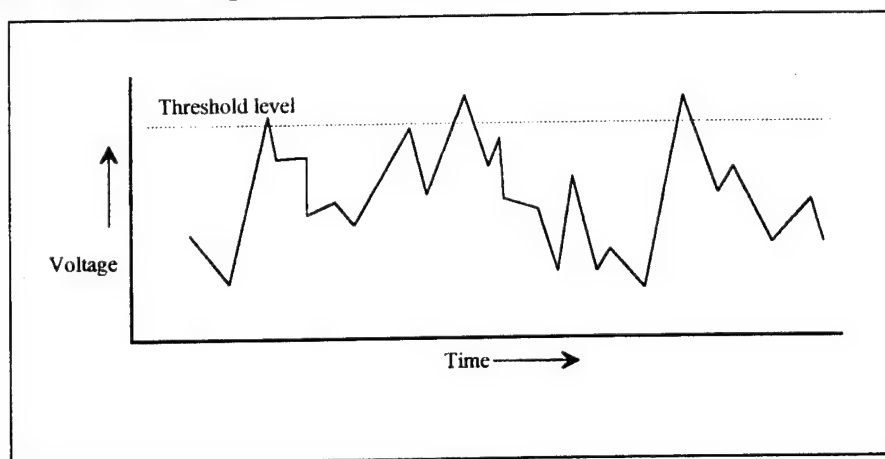


Figure 1.

Typical Envelope of a Radar Receiver Output as a Function of Time. (After Skolnik, 1980)

A low threshold increases the chance that noise alone will rise above the threshold and generate *false alarms*. On the other hand, a high threshold will result in missed targets. Consequently, the selection of a proper threshold is a compromise between reducing the PFA and increasing the PD.

To make things more difficult, subtle changes in the threshold can drastically change the overall system PFA. Although these changes can be manually compensated by trained operators (indeed this is how old systems operated), current and future processors require near-instantaneous changes in threshold levels to maintain a constant PFA to prevent overloading automatic detection and tracking systems (Skolnik, 1980). One such processor is the GO CFAR or Greatest-Of, Constant False Alarm Rate device.

B. GO CFAR PROCESSOR

In the GO CFAR processor, the background noise or clutter in the vicinity of a target cell is sampled, then the threshold is adjusted to provide a constant false alarm rate. Figure 2 depicts a how a typical radar might view a target in a particular region in space.

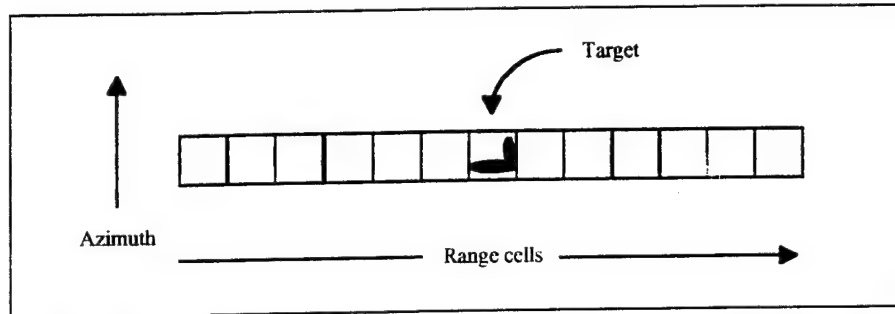
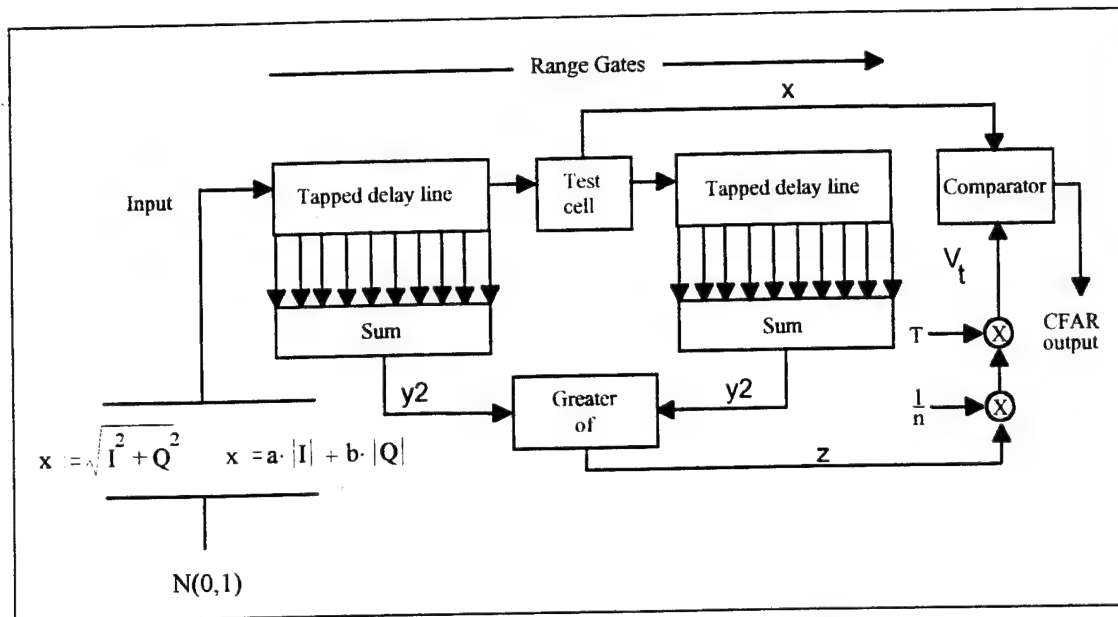


Figure 2. Typical Radar View of Target Region.

For sake of simplicity, assume that a target is located completely within a single range cell, while the remaining cells contain only thermal noise. Once received, the signal is frequently manipulated to create a *quadrature* channel which is 90° out of phase with the in-phase signal. The purpose of the quadrature channel is to retain the phase information and eliminate the effects of *blind phases* (Skolnik, 1980).

A schematic diagram of the GO CFAR processor is shown in Figure 3. The input I / Q thermal noise samples are considered to be normally distributed $N(0,1)$ and detected as either $x = \sqrt{I^2 + Q^2}$ or $x = a|I| + b|Q|$ in the test and reference cells. Both reference cell neighborhoods contain n cells which are used to determine the noise power levels y_1, y_2 . The detector threshold voltage V_t is obtained by choosing the *greater of* y_1 and y_2 , normalizing by the number of reference cells n , and multiplying by the threshold multiplier T . A target is declared if the amplitude of the test cell is greater than V_t .



The threshold multiplier (T) is normally set *a priori* to achieve a desired false alarm rate (Pace, 1994). For any fixed T , increasing the number of cells sampled (n) corresponds with a decrease in the false alarm rate. The probability of detection, on the other hand, is based upon the SNR of the received signal and is normally outside the realm of the designer's control.

For mathematical convenience, CFAR processors are frequently analyzed using the envelope detector $x_e = \sqrt{I^2 + Q^2}$ to detect the In-phase (I) and Quadrature (Q) channels of the baseband signal. To reduce hardware complexities created by the use of squares and square roots in the envelope detector, numerous envelope approximations have been developed in the form $x = a|I| + b|Q|$ where a and b are simple multiplying coefficients (Pace, 1994). These simplifications result in a slight loss in sensitivity, a difference in PFA performance (Pace, 1994) and consequently, a decrease in detection probability. This study analyzes the detector performance of the GO CFAR processor using the envelope detector and several envelope detector approximations. Since numerical integration of the PD expressions require considerable time, Monte Carlo

simulations are used to obtain the results. Closed form expressions for the PD are then found using curve fitting techniques.

C. NOISE AND PROBABILITY

Radar signal analysis requires an initial understanding of noise which is a random phenomenon.

1. Probability Density Functions

Many sources of noise can be modeled by a gaussian or normal density function, also commonly referred to as the *bell curve* as illustrated below in Figure 4,

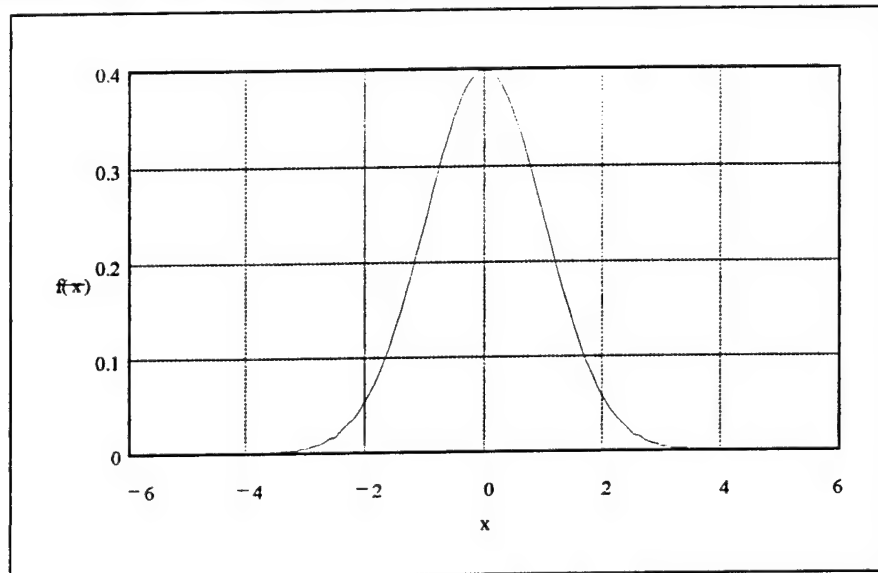


Figure 4. Gaussian, Normal or Bell Curve

and given by

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\eta)^2}{2\sigma^2}} \quad (3.1)$$

where η =mean and σ = standard deviation, $N(\eta,\sigma)$. The x-axis represents signal amplitude, the y-axis the probability of a signal associated with that amplitude. Using this model, noise is comprised primarily of low amplitude noise centered around zero,

whereas the probability of higher amplitude noise drops off exponentially with increasing amplitude.

2. Cumulative Distribution Functions

The cumulative distribution function (cdf) is the probability that the value x is less than some specified value and is given mathematically as the integral of the density function

$$F(y) = \int_{-\infty}^y f(x) dx. \quad (3.2)$$

By convention, cdfs are represented by capital letters, while pdfs are represented by lower-case letters. The cdf of the density function in Figure 4 appears in Figure 5.

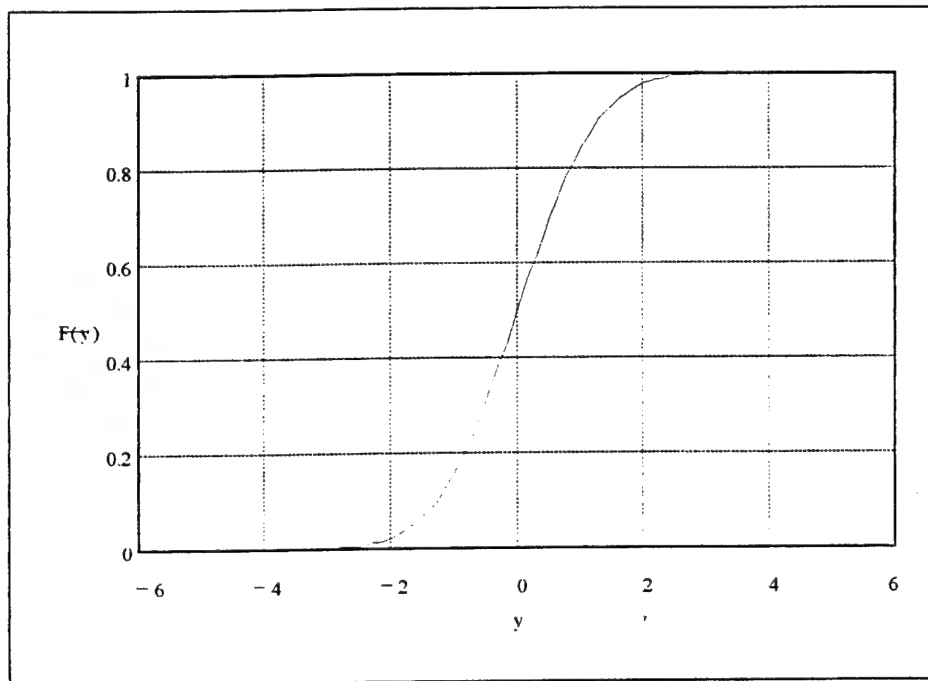


Figure 5. Typical Cumulative Distribution Function.

3. Summation of Density Functions

Given two different functions of similar, but independent random variables the pdf of the output is equal to the *convolution* of the density functions (Papoulis, 1984):

$$f_z(z) = \int_{-\infty}^{\infty} f_x(z-y)f_y(y)dy \quad (3.3)$$

Note that if $f_x(x) = 0$ for $x < 0$ and $f_y(y) = 0$ for $y < 0$, then $f_z(z) = 0$ for $z < 0$ and

$$f_z(z) = \int_0^z f_x(z-y)f_y(y)dy \quad z > 0. \quad (3.4)$$

4. Maximum of Two Density Function

Given two functions of independent random variables $f_x(x)$ and $f_y(y)$, cdf of the *maximum* of the two functions is given as (Papoulis, 1984):

$$F_z(z) = F_x(z)F_y(z) \quad (3.5)$$

and the corresponding pdf is given as:

$$f_z(z) = f_x(z)F_y(z) + f_y(z)F_x(z). \quad (3.6)$$

If the pdfs of $f_x(x)$ and $f_y(y)$ are equivalent, then the pdf simplifies to

$$f_z(z) = 2f_x(z)F_x(z) = 2f_y(z)F_y(z). \quad (3.7)$$

5. Linear transformation of density functions

For the linear transformation $y=ax$ where x is a random variable with pdf $f_x(x)$, the pdf of y , $f_y(y)$ is (Papoulis, 1984):

$$f_y(y) = \frac{1}{|a|}f_x\left(\frac{y}{a}\right). \quad (3.8)$$

6. Absolute value of a density function

For the absolute value of a function of a random variable $y=|x|$, the resulting pdf is given as

$$f_y(y) = f_x(y) + f_x(-y). \quad (3.9)$$

The relationships discussed above are needed to determine the density function for the detector approximation being examined.

D. ENVELOPE DETECTOR AND APPROXIMATIONS

The input to the GO CFAR processor is composed of a combination of in-phase and quadrature signals which are 90° out of phase and of the form $I = A \cos(\phi) + x$ and $Q = A \sin(\phi) + y$ where A is the signal amplitude, ϕ is the signal phase and x, y are Gaussian random variables with zero mean and variance of one $N(0,1)$. The associated pdfs of the absolute value of these two signals and associated curves, with $\phi = \pi/8$ are shown below:

$$f_{|I|}(x, A, \phi) = \frac{e^{-\frac{(x-A \cos(\phi))^2}{2}} + e^{-\frac{(-x-A \cos(\phi))^2}{2}}}{\sqrt{2\pi}} \quad x > 0 \quad (3.10)$$

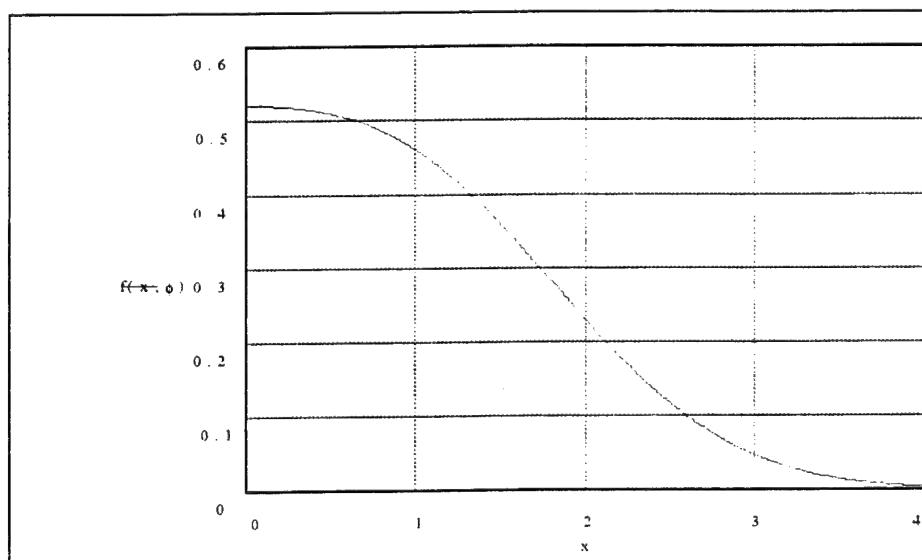


Figure 6. PDF of $|I|$ with $\phi = \pi/8$

$$f_{|Q|}(x, A, \phi) = \frac{e^{-\frac{(x-A \sin(\phi))^2}{2}} + e^{-\frac{(-x-A \sin(\phi))^2}{2}}}{\sqrt{2\pi}} \quad x > 0 \quad (3.11)$$

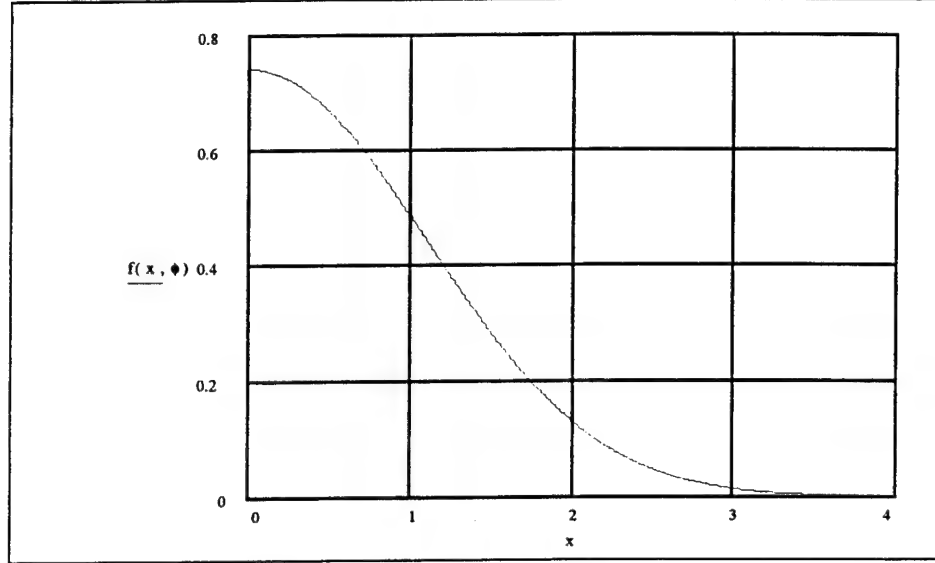


Figure 7. PDF of $|Q|$ with $\phi = \pi/8$

Using Equation (3.8) to factor in the a and b coefficients for the envelope approximation, we obtain the following pdfs:

$$f_{a|I|}(x, \phi) = \frac{e^{-\frac{\left(\frac{x}{a} - A \cos(\phi)\right)^2}{2}} + e^{-\frac{\left(-\frac{x}{a} - A \cos(\phi)\right)^2}{2}}}{|a|\sqrt{2\pi}} \quad x > 0 \quad (3.12)$$

$$f_{b|Q|}(y, \phi) = \frac{e^{-\frac{\left(\frac{y}{b} - A \sin(\phi)\right)^2}{2}} + e^{-\frac{\left(-\frac{y}{b} - A \sin(\phi)\right)^2}{2}}}{|b|\sqrt{2\pi}} \quad x > 0 \quad (3.13)$$

As previously discussed, the envelope detector is given by $x_e = \sqrt{I^2 + Q^2}$ while the approximation formula is $x = a|I| + b|Q|$. Seven sets of multiplying coefficients for a and b which provide reasonable approximations to the envelope detector are given in Table 1 below.

| Approximation | a | b |
|---------------|---------|---------|
| 1 | 1 | 1 |
| 2 | 1 | 1/2 |
| 3 | 1 | 1/4 |
| 4 | 1 | 3/8 |
| 5 | 31/32 | 3/8 |
| 6 | 0.948 | 0.393 |
| 7 | 0.96043 | 0.39782 |

Table 1.

Multiplying Coefficients a and b for Seven Envelope Approximations (After Filip, 1976)

E. PROBABILITY OF FALSE ALARM VS THRESHOLD MULTIPLIER

Using the results of previous sections, the PFA of the GO CFAR processor is shown. The GO CFAR processor is reproduced in the figure below.

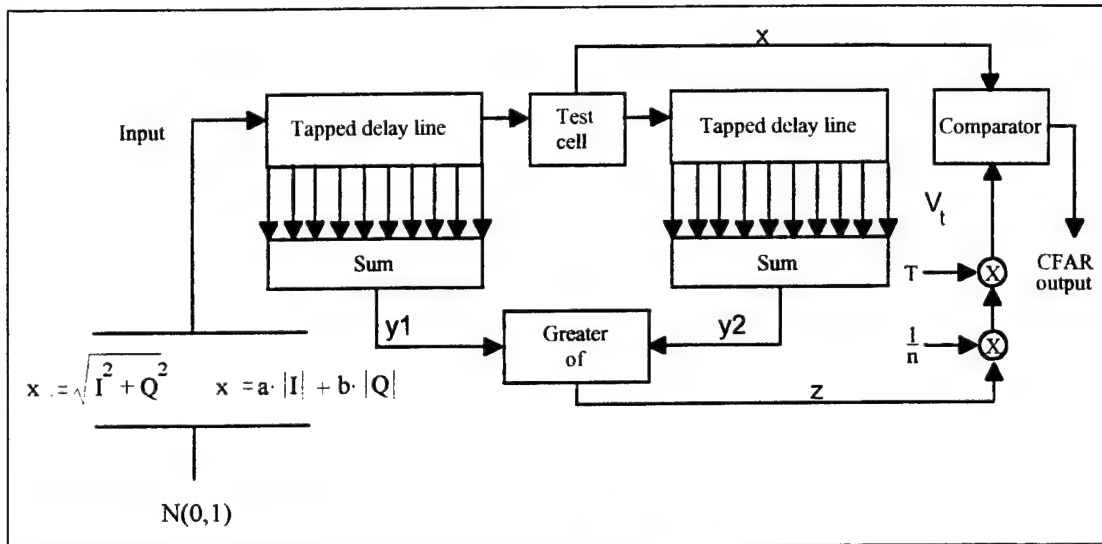


Figure 8. GO CFAR (After Skolnik, 1980)

Since we're interested in the PFA, we start with the assumption that no target is present in the test cell and ask ourselves: What is the probability that the signal Tz/n is higher than the test cell output x ? The answer to this question gives us the probability of false alarm. We start with the pdf of the input signal, step through the GO CFAR model calculating corresponding pdfs at each junction, and end up with the PFA at the output of the GO CFAR.

With no target present ($A=0$), the pdfs of $|I|$ and $|Q|$ are greatly simplified. In fact, they are the same.

$$f_{|I|}(x) = f_{|Q|} = \sqrt{\frac{2}{\pi}} \left(e^{-\frac{x^2}{2}} \right) \quad (3.14)$$

For the envelope approximation $x = a|I| + b|Q|$, we use our previous results to calculate the pdf of the input as the *convolution* of the pdfs of $a|I|$ and $b|Q|$. The resultant pdf becomes

$$f_x(x) = \sqrt{\frac{2}{\pi(a^2+b^2)}} e^{\frac{-x^2}{2(a^2+b^2)}} \left(\operatorname{erf}\left(\frac{a}{b} \frac{x}{\sqrt{2(a^2+b^2)}}\right) + \operatorname{erf}\left(\frac{b}{a} \frac{x}{\sqrt{2(a^2+b^2)}}\right) \right) \quad (3.15)$$

where $\operatorname{erf}(x)$ is the error function given by

$$\operatorname{erf}(x) = \frac{1}{\sqrt{2\pi}} \int_0^x e^{-\frac{y^2}{2}} dy. \quad (3.16)$$

Assuming the input contains only noise with no target present, Equation (3.16) also represents the pdf of a test cell output.

The summation of input signals which result in the outputs y_1 and y_2 are a function of the number of reference cells n . Using the results of Equation (3.4), we note that $f_{y_1}(y_1)$ and $f_{y_2}(y_2)$ are both n -fold convolutions of $f_x(x)$ for which there is no closed-form solution.

The pdf of z after the *Greater-Of* block can be determined using Equation (3.7):

$$f_z(z) = 2f_{y_1}(z)F_{y_1}(z) = 2f_{y_2}(z)F_{y_2}(z) \quad (3.17)$$

where $F_{y_1}(z)$ is the cumulative distribution function of $f_{y_1}(z)$. The PFA at the comparator output is then

$$PFA(T) = \int_0^\infty f_z(z) \left\{ \int_{Tz/n}^\infty f_x(\lambda) d\lambda \right\} dz. \quad (3.18)$$

Substituting in Equation 3.18 and integrating F_{y1} we get

$$PFA(T) = 2 \int_0^\infty f_{y1}(z) \left\{ \int_0^z f_{y1}(\xi) d\xi \right\} \left[\int_{Tz/n}^\infty f_x(\lambda) d\lambda \right] dz \quad (3.19)$$

where $f_x(\lambda)$ is given Equation (3.15). Although this equation has no closed form solution, it can be calculated numerically. Also, closed form expressions for the PFA are given in Pace. These results are used to extract the corresponding threshold multipliers used in the Monte Carlo simulations for the detection performance.

F. PROBABILITY OF DETECTION VS. SNR

The determination of the probability of detection follows a similar line of analysis as in the previous section. We now assume however, a target is present in the test cell. In addition, the threshold multiplier T is no longer a variable, but a constant which corresponds with a particular PFA. This time, we ask the question: What is the probability that the signal from the test cell containing the target is higher than the signal Tz/n appearing at the comparator? The answer to this question gives us the probability of detection.

Our first step is to calculate the pdf of the input signal x using the envelope approximation. Using Equation (3.4) once again, we can calculate the pdf of the envelope approximation as the *convolution* of the pdfs of $a|I|$ and $b|Q|$ with a target present (Equations (3.12) and (3.13) respectively):

$$f_x(x, A, \phi) = \int_0^x \frac{e^{-\frac{\left(\frac{x-\lambda}{a} - A \cos(\phi)\right)^2}{2}} + e^{-\frac{\left(-\frac{x-\lambda}{a} - A \cos(\phi)\right)^2}{2}}}{|a|\sqrt{2\pi}} \times \frac{e^{-\frac{\left(\frac{\lambda}{b} - A \sin(\phi)\right)^2}{2}} + e^{-\frac{\left(-\frac{\lambda}{b} - A \sin(\phi)\right)^2}{2}}}{|b|\sqrt{2\pi}} d\lambda \quad (3.20)$$

Substituting in the following relation of SNR to signal amplitude A (Wilson, 1982),

$$SNR = \frac{A^2}{2} \quad (3.21)$$

and performing the integration, we obtain the following closed-form solution.

$$f_x(x, SNR, \phi) = A \left[\begin{aligned} &e^{\left(\frac{-(B-x)^2}{2D}\right)} \left(\operatorname{erf}\left(\frac{F-G+J}{a\sqrt{2D}}\right) - \operatorname{erf}\left(\frac{-E+H-I}{b\sqrt{2D}}\right) \right) + \\ &e^{\left(\frac{-(B+x)^2}{2D}\right)} \left(-\operatorname{erf}\left(\frac{F-G-J}{a\sqrt{2D}}\right) + \operatorname{erf}\left(\frac{-E+H+I}{b\sqrt{2D}}\right) \right) + \\ &e^{\left(\frac{-(C-x)^2}{2D}\right)} \left(-\operatorname{erf}\left(\frac{F+G-J}{a\sqrt{2D}}\right) + \operatorname{erf}\left(\frac{E+H-I}{b\sqrt{2D}}\right) \right) + \\ &e^{\left(\frac{-(C+x)^2}{2D}\right)} \left(\operatorname{erf}\left(\frac{F+G+J}{a\sqrt{2D}}\right) - \operatorname{erf}\left(\frac{E+H+I}{b\sqrt{2D}}\right) \right) + \end{aligned} \right] \quad (3.22)$$

with the following substitutions:

$$A = \frac{ab}{2|a||b|\sqrt{2\pi(a^2+b^2)}} \quad (3.23)$$

$$B = \sqrt{2SNR} (a \cos(\phi) + b \sin(\phi)) \quad (3.24)$$

$$C = \sqrt{2SNR} (a \cos(\phi) - b \sin(\phi)) \quad (3.25)$$

$$D = a^2 + b^2 \quad (3.26)$$

$$E = \sqrt{2SNR} b^2 \cos(\phi) \quad (3.27)$$

$$F = \sqrt{2SNR} a^2 \sin(\phi) \quad (3.28)$$

$$G = \sqrt{2SNR} ab \cos(\phi) \quad (3.29)$$

$$H = \sqrt{2SNR} ab \sin(\phi) \quad (3.30)$$

$$I = ax \quad (3.31)$$

$$J = bx \quad (3.32)$$

The PD equation is similar in form to the PFA equation, Equation (3.19), but T is now constant and derived for a specific PFA.

$$PD(SNR, \phi) = \int_0^\infty 2f_{y1}(z) \left\{ \int_0^z f_{y1}(\xi) d\xi \right\} \left[\int_{Tz/n}^\infty f_x(\lambda, SNR, \phi) d\lambda \right] dz \quad (3.33)$$

Since the phase of the target signal is also random, the PD must be averaged over the phase as

$$PD(SNR) = \int_0^{\pi/2} \frac{2}{\pi} \int_0^\infty 2f_{y1}(z) \left\{ \int_0^z f_{y1}(\xi) d\xi \right\} \left[\int_{Tz/n}^\infty f_x(\phi, SNR, \lambda) d\lambda \right] dz d\phi. \quad (3.34)$$

Once again, there is no closed form expression for the PD equation, but it can be evaluated using numerical integration or Monte Carlo simulations. The latter approach is the subject of the following chapter. This is necessary in order to develop the closed form expressions to approximate this detection performance.

III. MONTE CARLO SIMULATIONS USING MATLAB

A. MATLAB PROGRAM

Monte Carlo simulations are conducted using the MATLAB code located in the appendix. The goal is to obtain the PD vs. SNR curves associated with a PFA value of 10^{-4} for each envelope approximation. The PFA chosen is a value generally reported in the literature for CFAR comparisons. For both the envelope approximation and envelope detector, the following number of reference cells $n = 1, 2, 4, 8, 16, 32$ are used. Threshold multipliers (T) for each particular envelope approximation, the number of reference cells (n) and the desired PFA, are obtained by interpolating the PFA vs. the threshold multiplier data from a previous study (Pace, 1994). The tables below summarize the threshold multipliers used for each Monte Carlo simulation.

| n | T for $PFA=10^{-4}$ |
|-----|-----------------------|
| 1 | 12.287 |
| 2 | 5.952 |
| 4 | 4.423 |
| 8 | 3.891 |
| 16 | 3.68 |
| 32 | 3.5975 |

Table 2. Threshold Values for Envelope Approximation $a=1$, $b=1$

| n | T for $PFA=10^{-4}$ |
|-----|-----------------------|
| 1 | 13.43 |
| 2 | 6.515 |
| 4 | 4.773 |
| 8 | 4.1525 |
| 16 | 3.903 |
| 32 | 3.804 |

Table 3. Threshold Values for Envelope Approximation $a=1$, $b=1/2$

| n | T for PFA= 10^{-4} |
|----|----------------------|
| 1 | 16.685 |
| 2 | 7.915 |
| 4 | 5.53 |
| 8 | 4.675 |
| 16 | 4.335 |
| 32 | 4.197 |

Table 4. Threshold Values for Envelope Approximation $a=1$, $b=1/4$

| n | T for PFA= 10^{-4} |
|----|----------------------|
| 1 | 14.53 |
| 2 | 7.018 |
| 4 | 5.064 |
| 8 | 4.361 |
| 16 | 4.076 |
| 32 | 3.9636 |

Table 5. Threshold Values for Envelope Approximation $a=1$, $b=3/8$

| n | T for PFA= 10^{-4} |
|----|----------------------|
| 1 | 14.399 |
| 2 | 6.96 |
| 4 | 5.032 |
| 8 | 4.338 |
| 16 | 4.058 |
| 32 | 3.9465 |

Table 6. Threshold Values for Envelope Approximation $a=31/32$, $b=3/8$

| n | T for PFA= 10^{-4} |
|----|----------------------|
| 1 | 14.112 |
| 2 | 6.835 |
| 4 | 4.96 |
| 8 | 4.287 |
| 16 | 4.017 |
| 32 | 3.907 |

Table 7. Threshold Values for Envelope Approximation $a=0.948$, $b=0.393$

| n | T for PFA=10 ⁻⁴ |
|----|----------------------------|
| 1 | 14.114 |
| 2 | 6.834 |
| 4 | 4.96 |
| 8 | 4.288 |
| 16 | 4.017 |
| 32 | 3.908 |

Table 8. Threshold Values for Envelope Approximation $a=0.96043$, $b=0.39782$

| n | T for PFA=10 ⁻⁴ |
|----|----------------------------|
| 1 | 11.962 |
| 2 | 5.751 |
| 4 | 4.251 |
| 8 | 3.729 |
| 16 | 3.5215 |
| 32 | 3.4395 |

Table 9. Threshold Values for Envelope Detector $\sqrt{I^2 + Q^2}$

B. PROGRAMMING TECHNIQUES

To simulate noise-only inputs to the reference cells, the program creates matrices of random numbers $N(0,1)$ with dimensions $(n \times N)$, where n = number of reference cells per side and N = number of simulations desired. The absolute values of these matrices are summed across the n variable to create a vector of dimension $(1 \times N)$, then multiplied by the appropriate coefficients a or b and added together. This creates two vectors of dimension $(1 \times N)$ which represent the summation of reference cells on the left and right side of the test cell. These two vectors are compared, and for each N , the *greater of* the two cells is placed into a new vector z of dimension $(1 \times N)$. To create the comparison voltage V_t we divide by the number of reference cells n , and multiply by the threshold multiplier T . The test cell signal, V_{test} , is created in a similar manner, but this time we add a signal with amplitude A which corresponds to the SNR to be tested.

The next step compares the two vectors V_t and V_{test} and calculates the percentage of times that $V_{test} > V_t$. For each SNR tested, the process is repeated for

$0 < \phi < \pi/2$ since the phase of noise is equally distributed. The result is a probability of detection for a single SNR. This must be repeated for the entire range of SNRs to be tested.

After a number of trial runs, it was determined that 400,000 simulations per SNR with an interval of 0.5 dB provided reasonably *smooth* curves for our analysis. We also took advantage of the fact that at some point, when the SNR becomes large enough, the PD reaches 1.0. Thus, the algorithm breaks the loop after five successive PDs of 1.0 to remove unnecessary computational loops. The PD plots are given below.

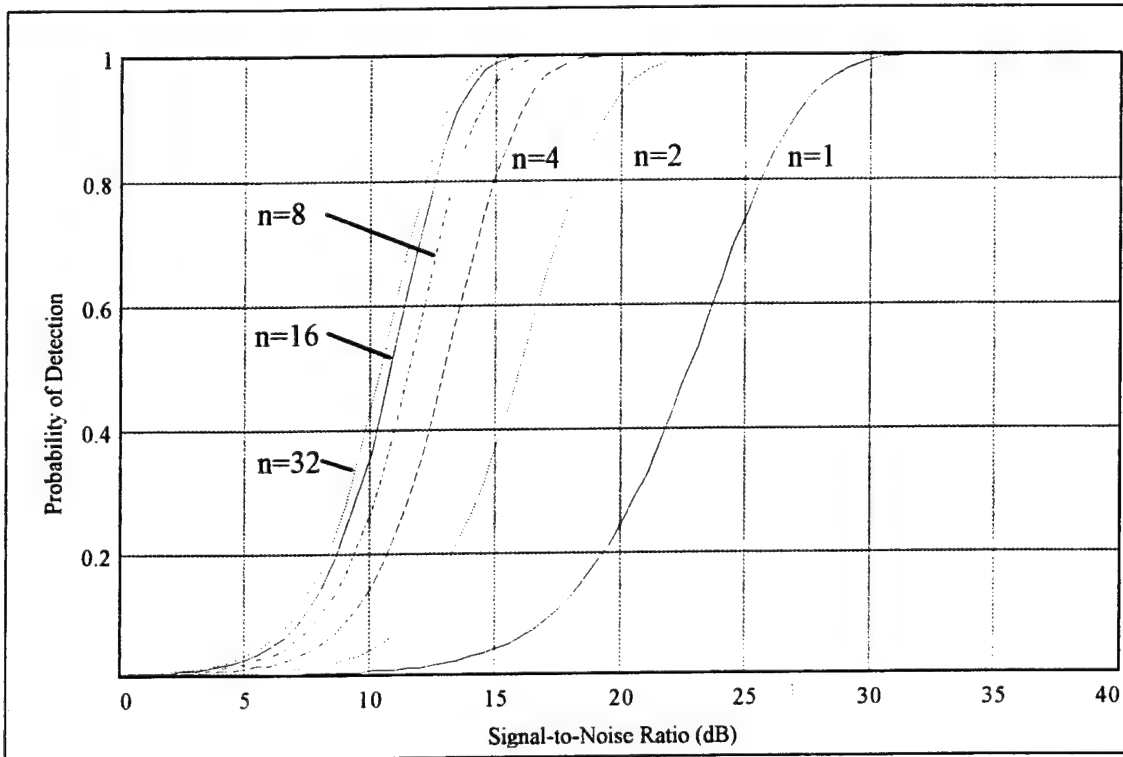


Figure 9: PD vs. SNR for $a=1$, $b=1$, $PFA=10^{-4}$

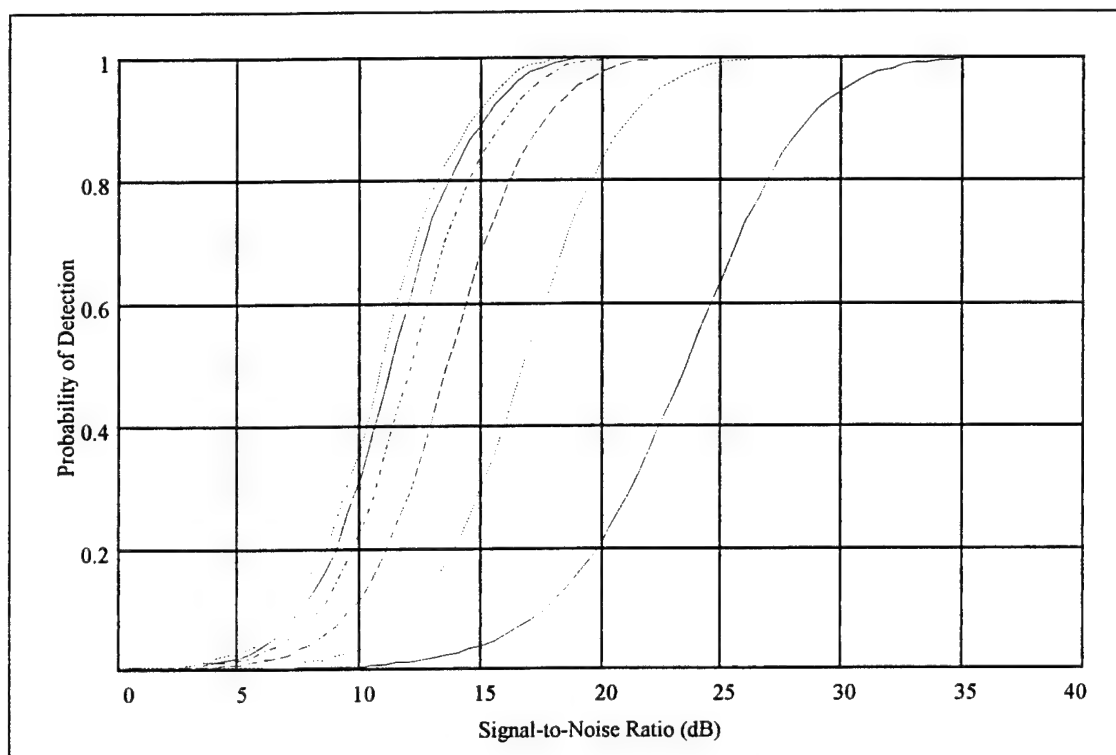


Figure 10. PD vs. SNR using Envelope Approximation, $PFA=10^{-4}$, $a=1$, $b=1/2$

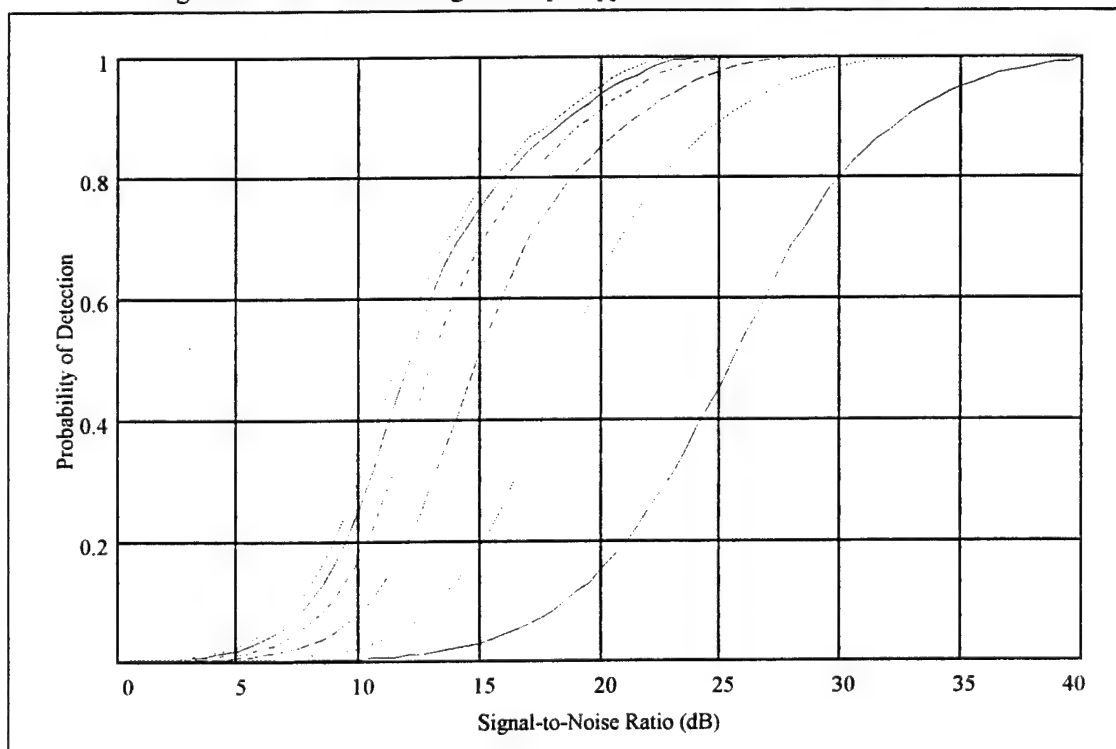


Figure 11. PD vs. SNR using Envelope Approximation, $PFA=10^{-4}$, $a=1$, $b=1/4$

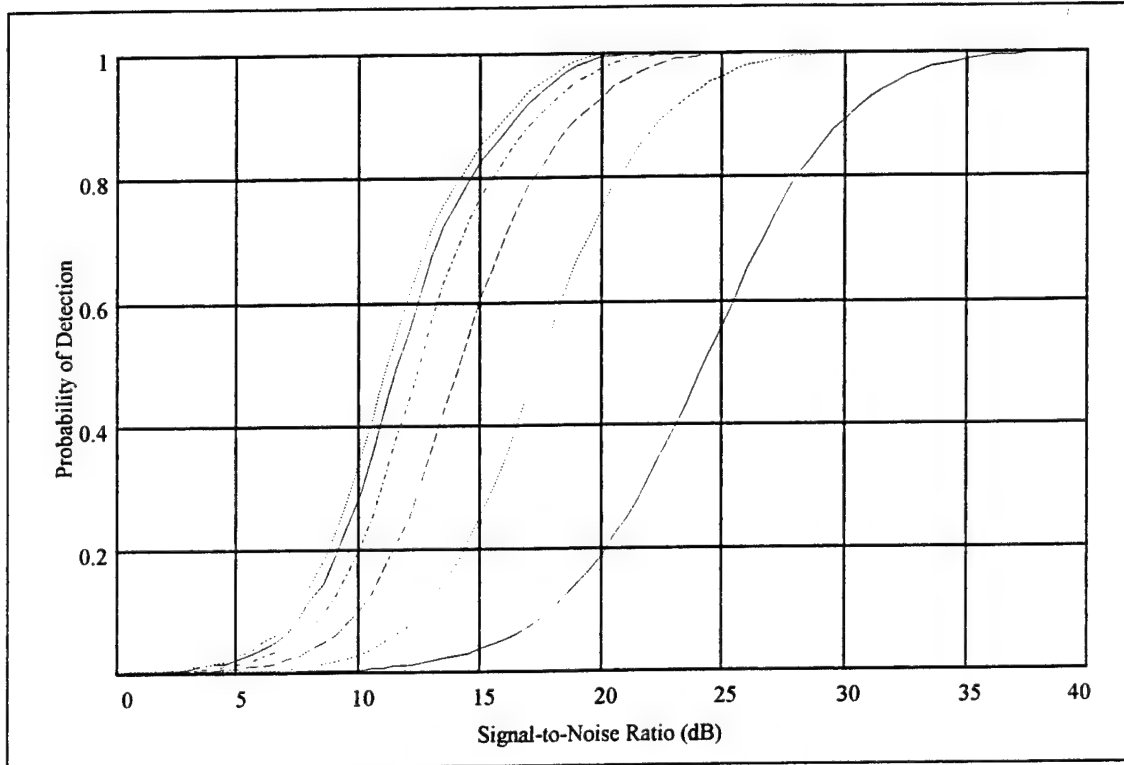


Figure 12. PD vs. SNR using Envelope Approximation, $PFA=10^{-4}$, $a=1$, $b=3/8$

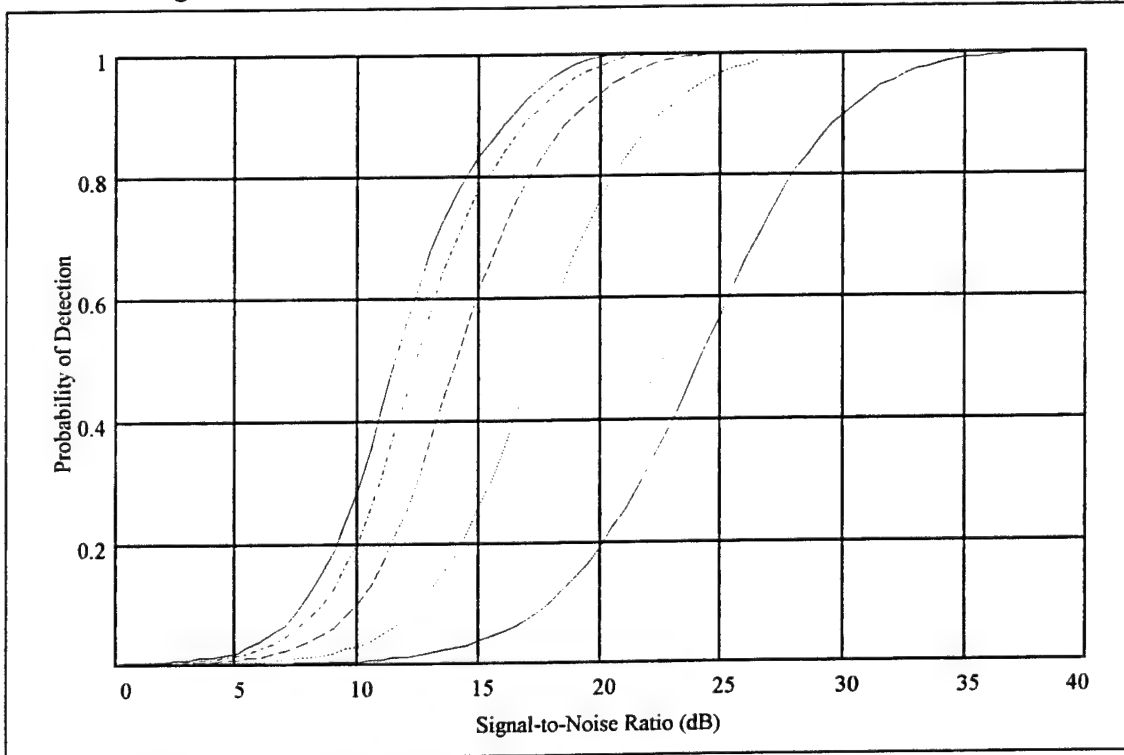


Figure 13. PD vs. SNR using Envelope Approximation, $PFA=10^{-4}$, $a=31/32$, $b=3/8$

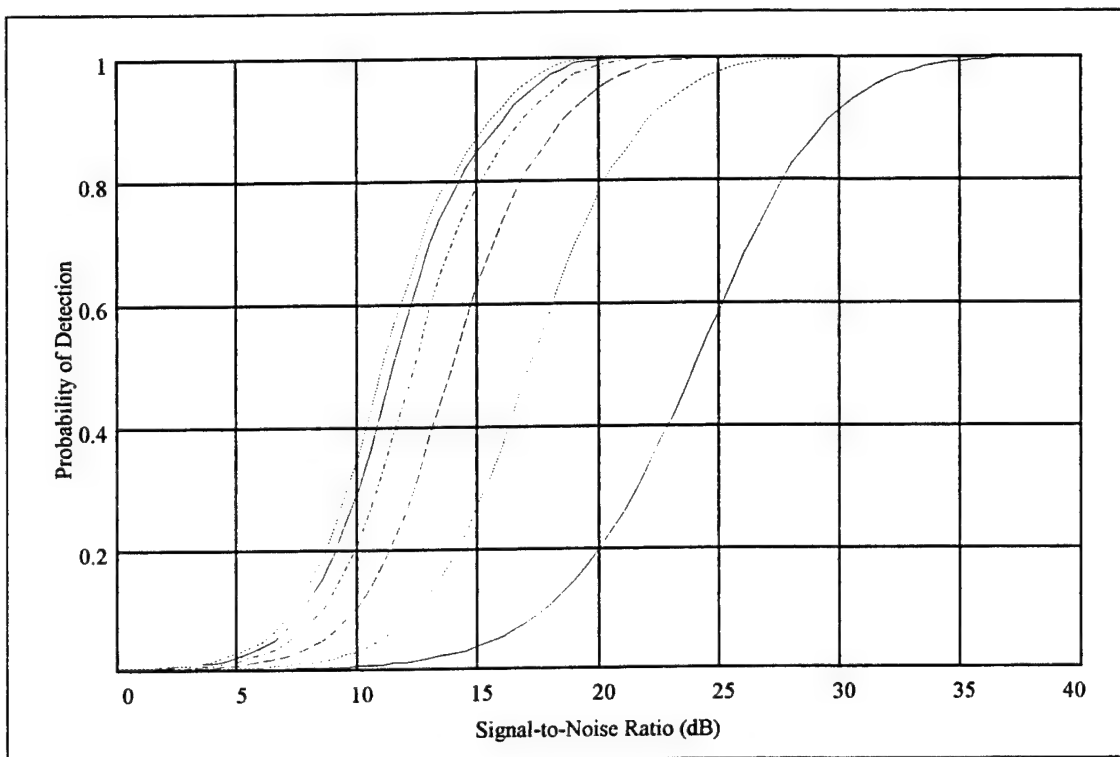


Figure 14. PD vs. SNR using Envelope Approximation, $PFA=10^{-4}$, $a=0.948$, $b=0.393$

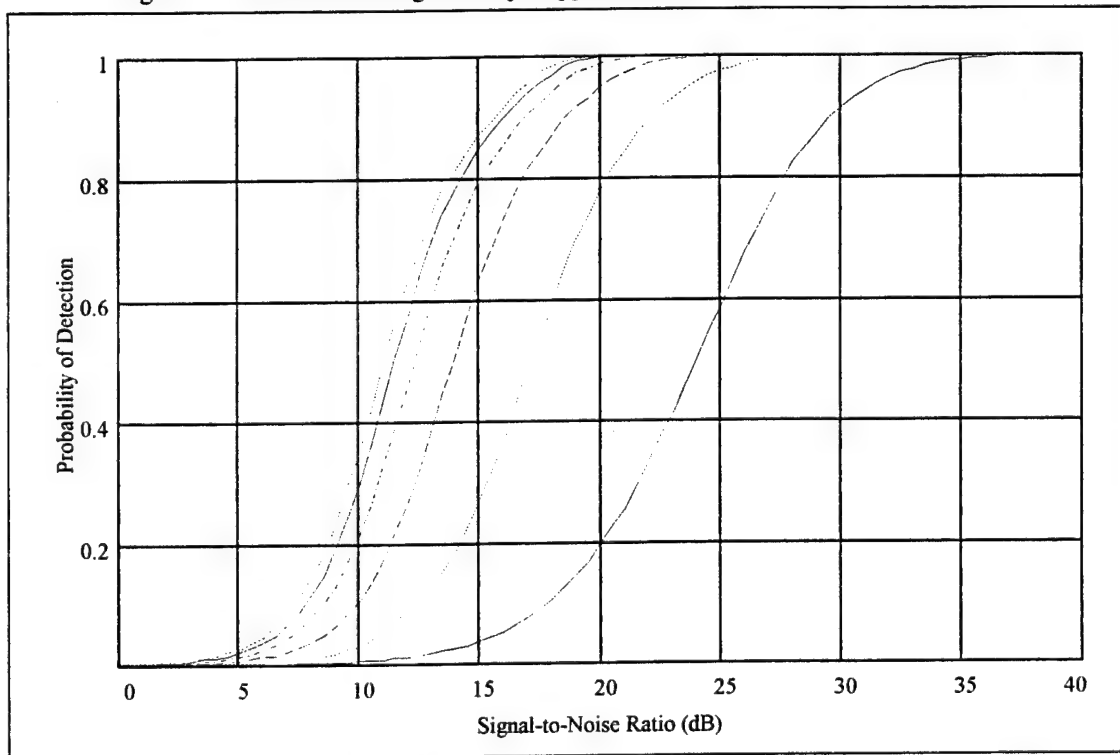


Figure 15. PD vs. SNR using Envelope Approximation, $PFA=10^{-4}$, $a=0.96043$, $b=0.39782$

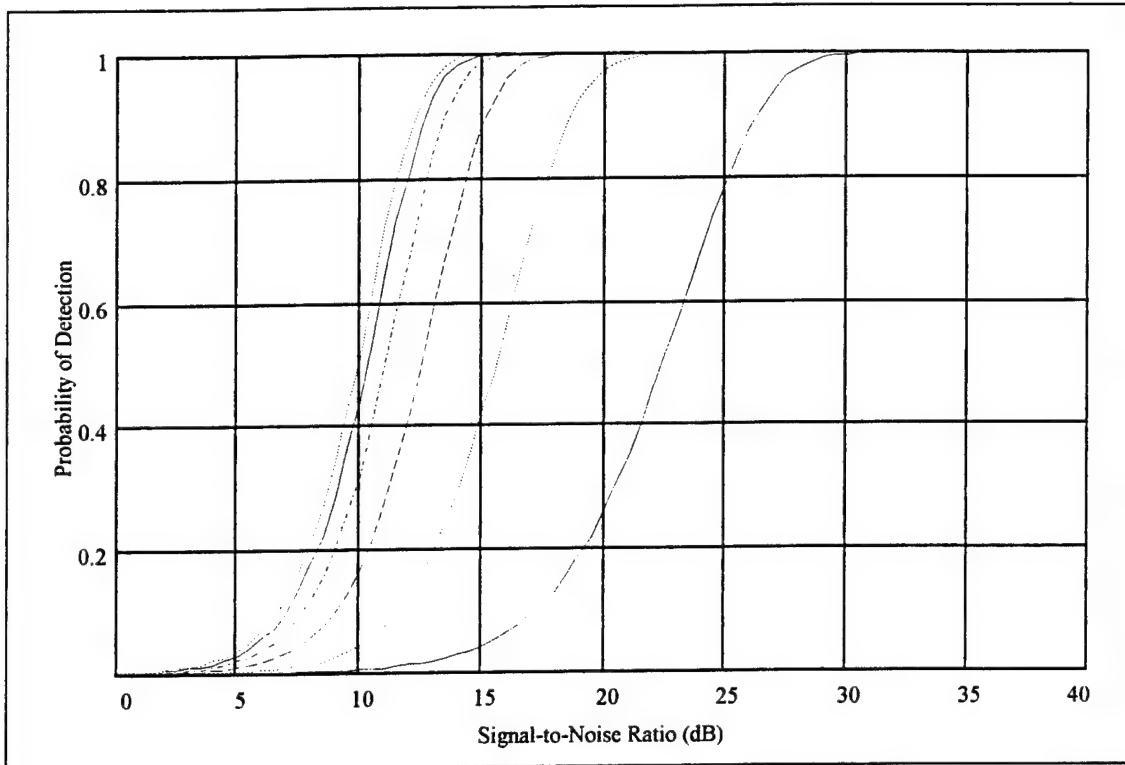


Figure 16. PD vs. SNR using Envelope Detector, $PFA=10^{-4}$

IV. APPROXIMATION FORMULAS FOR PD

This chapter outlines the derivation of closed-form, curve fit expressions for the various PD vs. SNR curves.

A. METHODS OF CURVE FITTING

Numerous methods were attempted to curve fit the Monte Carlo results and included the use of tanh, third-order exponentials, and *erf* functions. The most promising involved the use of the *erf* function.

B. ERF FUNCTION APPROXIMATION

The *erf* or error function, previously given in Equation (3.16), is repeated here along with a plot of the function for convenience

$$\text{erf}(x) = \frac{1}{\sqrt{2\pi}} \int_0^x e^{-\frac{y^2}{2}} dy \quad (4.1)$$

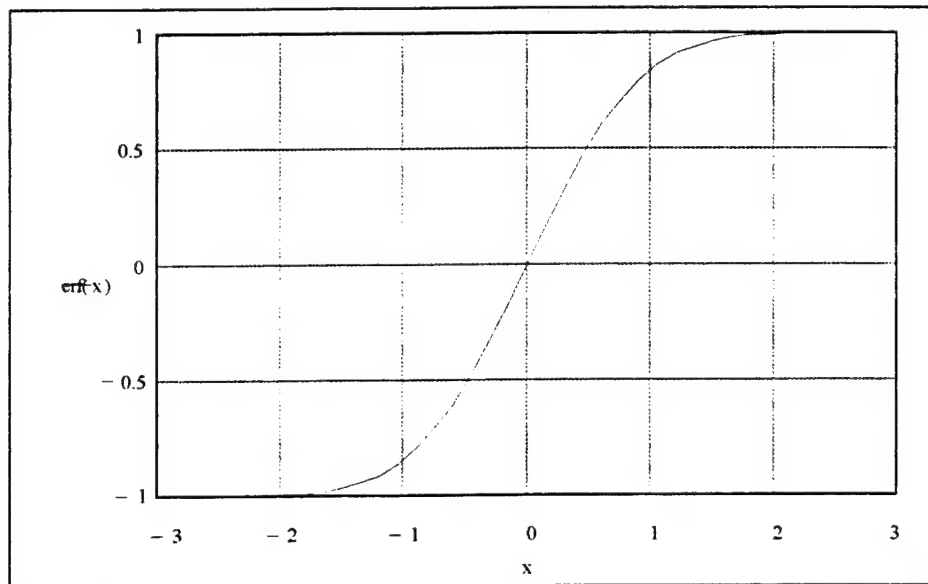


Figure 17: Plot of Erf Function

The *erf* function, due to its similar appearance to the PD curves in the previous section, is a natural candidate as a curve fit solution. By rewriting the function in the form

$$f(x) = 0.5 \left(\operatorname{erf} \left(\frac{x-c1}{c2} \right) + 1 \right) \quad (4.2)$$

we can control both the curvature and axes with the proper values of *c1* and *c2*. Using the PD vs. SNR curve for *n*=1 as an example, we can obtain a reasonably good fit using *c*=22.5 and *d*=5.26.

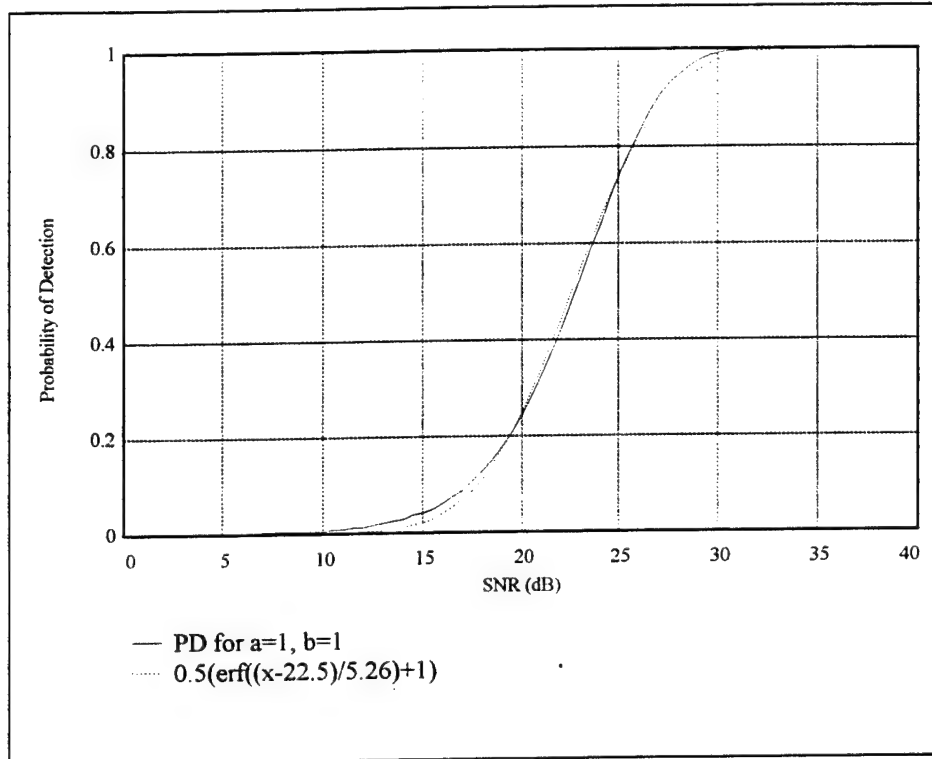


Figure 18: Comparison Between Actual PD Curve and Curve Fit Solution Using *Erf* Function

Despite the symmetric nature of the *erf* function, we are still able to produce a reasonable curve fit to the asymmetric solid PD curve (the line we are attempting to curve fit) and, at the same time, obtain a relatively low residual across the entire SNR axis. The residual is defined as

$$Residual(SNR) = |PD(SNR) - f(SNR)| \quad (4.3)$$

and describes the error or difference between the two curves. The residual plot is shown below.

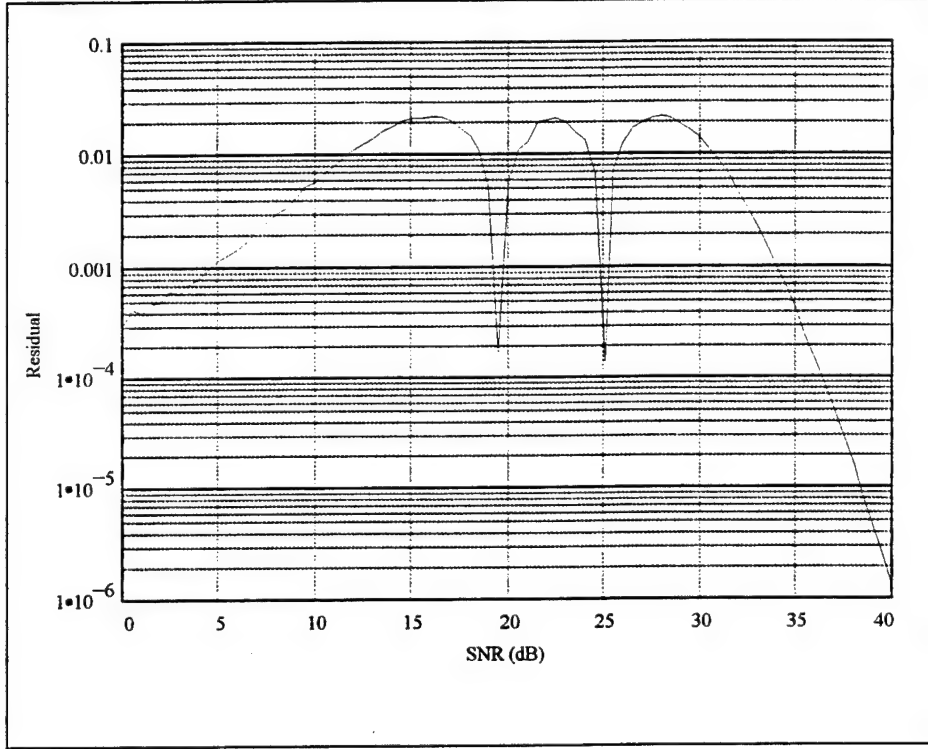


Figure 19: Residual of *Erf* Function Approximation

In this case, the maximum residual equals 0.0218. That is, for any given SNR, the *erf* approximation yields a PD result within 0.0218 of the actual value.

To improve this result, the curve-fit solution is split into two parts: PDs > 0.5 and PDs < 0.5. A curve fit solution is obtained for the *upper* portion of the curve above a PD of 0.5, and a second solution obtained for the *lower* curve. The formulas are given as:

$$PD < 0.5 \quad PD = 0.5\left(\operatorname{erf}\left(\frac{SNR-c1}{c2}\right) + 1\right) \quad (4.4)$$

$$PD > 0.5 \quad PD = 0.5\left(\operatorname{erf}\left(\frac{SNR-c1}{c3}\right) + 1\right) \quad (4.5)$$

where the coefficients $c1$, $c2$, $c3$ are determined experimentally.

In this case, the maximum error is reduced to 0.0116, clearly a better solution. Plots of the improved solution and corresponding residual curve for $a=1$, $b=1$, $n=1$, are given in the following figures.

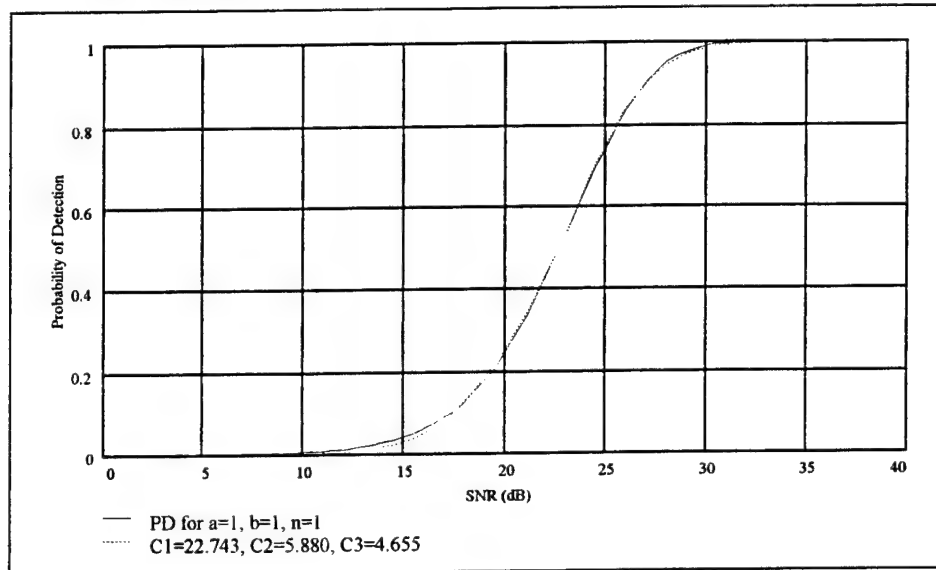


Figure 20: Comparison of Monte Carlo Simulation vs. Curve Fit Solution

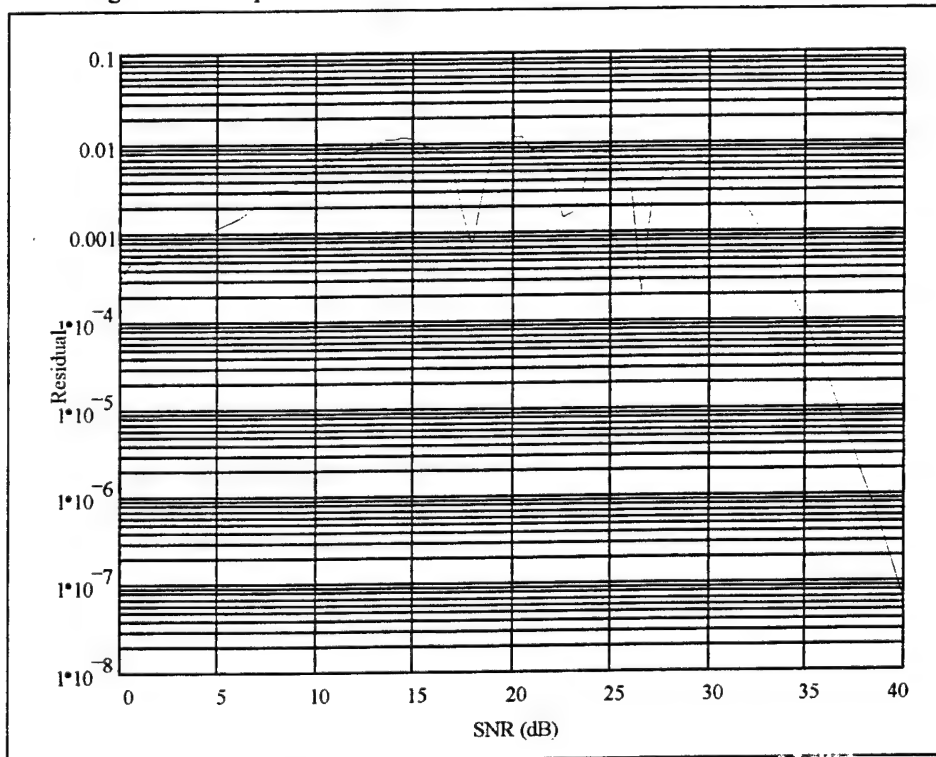


Figure 21: Residual of Approximation

C. TABLES OF COEFFICIENTS

The following tables provide the associated coefficients for each of the seven envelope approximations and envelope detector.

| n | c1 | c2 | c3 |
|----|--------|-------|-------|
| 1 | 22.743 | 5.88 | 4.655 |
| 2 | 15.958 | 4.494 | 3.726 |
| 4 | 12.964 | 3.93 | 3.229 |
| 8 | 11.546 | 3.807 | 2.928 |
| 16 | 10.8 | 3.677 | 2.832 |
| 32 | 10.384 | 3.602 | 2.866 |

Table 10. Coefficients for PD Curve $PFA=10^{-4}$, $a=1$, $b=1$

| n | c1 | c2 | c3 |
|----|--------|-------|-------|
| 1 | 22.52 | 5.703 | 6.421 |
| 2 | 16.8 | 4.67 | 5.018 |
| 4 | 13.59 | 4.362 | 4.268 |
| 8 | 12.082 | 4.066 | 4.104 |
| 16 | 11.27 | 4.015 | 3.919 |
| 32 | 10.869 | 4.037 | 3.948 |

Table 11. Coefficients for PD Curve $PFA=10^{-4}$, $a=1$, $b=1/2$

| n | c1 | c2 | c3 |
|----|--------|-------|-------|
| 1 | 25.569 | 7.636 | 7.597 |
| 2 | 18.521 | 5.815 | 6.878 |
| 4 | 14.919 | 4.98 | 6.531 |
| 8 | 13.006 | 4.595 | 6.552 |
| 16 | 12.078 | 4.489 | 6.366 |
| 32 | 11.631 | 4.552 | 6.356 |

Table 12. Coefficients for PD Curve $PFA=10^{-4}$, $a=1$, $b=1/4$

| n | c1 | c2 | c3 |
|----|--------|-------|-------|
| 1 | 24.267 | 6.901 | 6.426 |
| 2 | 17.433 | 5.311 | 5.591 |
| 4 | 14.165 | 4.643 | 5.136 |
| 8 | 12.468 | 4.34 | 5.008 |
| 16 | 11.573 | 4.191 | 5.041 |
| 32 | 11.132 | 4.14 | 5.002 |

Table 13. Coefficients for PD Curve $PFA=10^{-4}$, $a=1$, $b=3/8$

| n | c1 | c2 | c3 |
|----|--------|-------|-------|
| 1 | 24.173 | 6.825 | 6.351 |
| 2 | 17.334 | 5.231 | 5.523 |
| 4 | 14.072 | 4.595 | 5.091 |
| 8 | 12.373 | 4.196 | 4.989 |
| 16 | 11.545 | 4.179 | 4.963 |
| 32 | 11.072 | 4.006 | 4.976 |

Table 14. Coefficients for PD Curve $PFA=10^{-4}$, $a=31/32$, $b=3/8$

| n | c1 | c2 | c3 |
|----|--------|--------|-------|
| 1 | 23.982 | 6.7 | 6.163 |
| 2 | 17.165 | 5.156 | 5.339 |
| 4 | 13.958 | 4.491 | 4.849 |
| 8 | 12.319 | 4.264 | 4.694 |
| 16 | 11.488 | 4.187 | 4.597 |
| 32 | 11.025 | 4.1343 | 4.649 |

Table 15. Coefficients for PD Curve $PFA=10^{-4}$, $a=0.948$, $b=0.393$

| n | c1 | c2 | c3 |
|----|--------|-------|-------|
| 1 | 24 | 6.729 | 6.143 |
| 2 | 17.169 | 5.133 | 5.31 |
| 4 | 13.932 | 4.496 | 4.903 |
| 8 | 12.276 | 4.184 | 4.758 |
| 16 | 11.471 | 4.119 | 4.652 |
| 32 | 11.011 | 4.057 | 4.736 |

Table 16: Coefficients for PD Curve $PFA=10^{-4}$, $a=0.96043$, $b=0.39782$

| n | c1 | c2 | c3 |
|----|-------|-------|-------|
| 1 | 22.49 | 5.683 | 4.289 |
| 2 | 15.63 | 4.321 | 3.383 |
| 4 | 12.55 | 3.834 | 2.927 |
| 8 | 11.06 | 3.561 | 2.693 |
| 16 | 10.3 | 3.516 | 2.709 |
| 32 | 9.96 | 3.592 | 2.58 |

Table 17. Coefficients for PD Curve $PFA=10^{-4}$, Envelope Detector

V. PD APPROXIMATION COEFFICIENTS

A curve fit of the coefficients as a function of the number of reference cells n , allows system designers to extract the appropriate coefficients and obtain approximate PD vs. SNR curves for any number of reference cells n . The complete set of coefficient plots are given below.

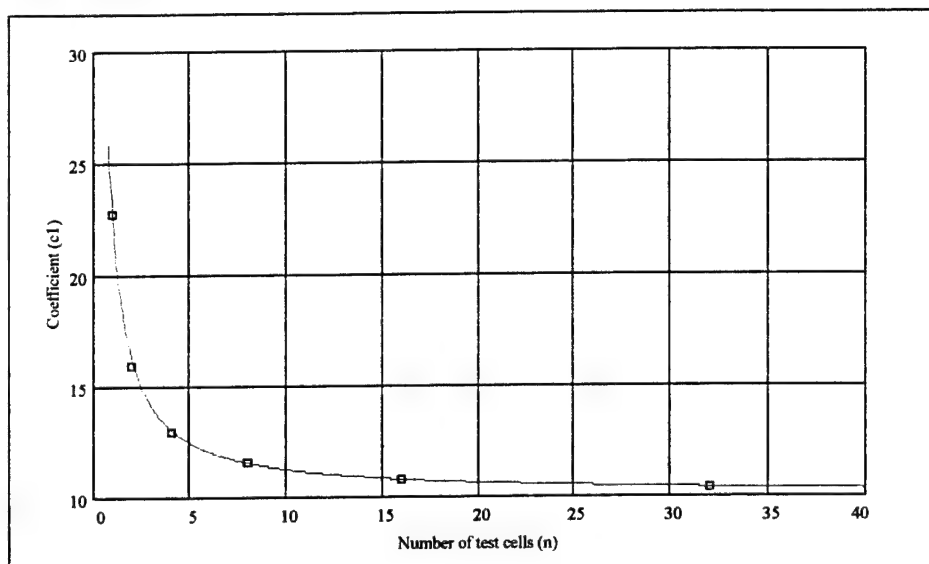


Figure 22: Coefficient C1 as a Function of n
For Envelope Approximation $a=1$, $b=1$

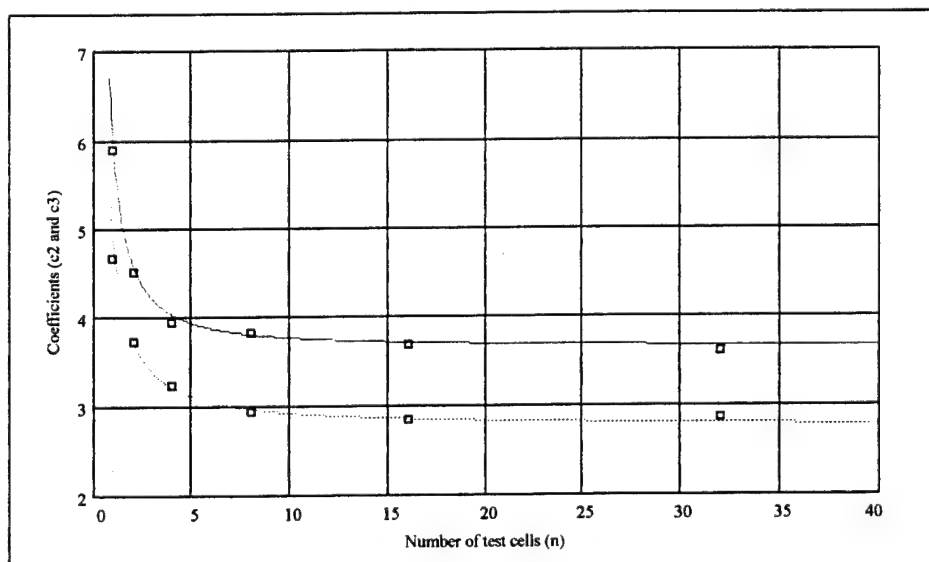


Figure 23: Coefficients C2 (Solid Line) and C3 (Dashed Line) as a Function of n
For Envelope Approximation $a=1$, $b=1$

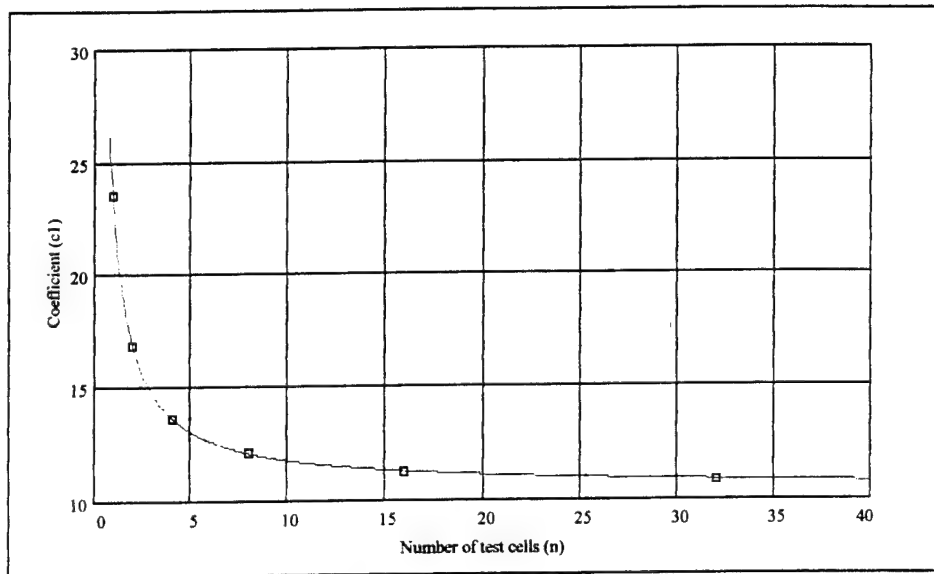


Figure 24: Coefficient C1 as a Function of n
For Envelope Approximation $a=1$, $b=1/2$

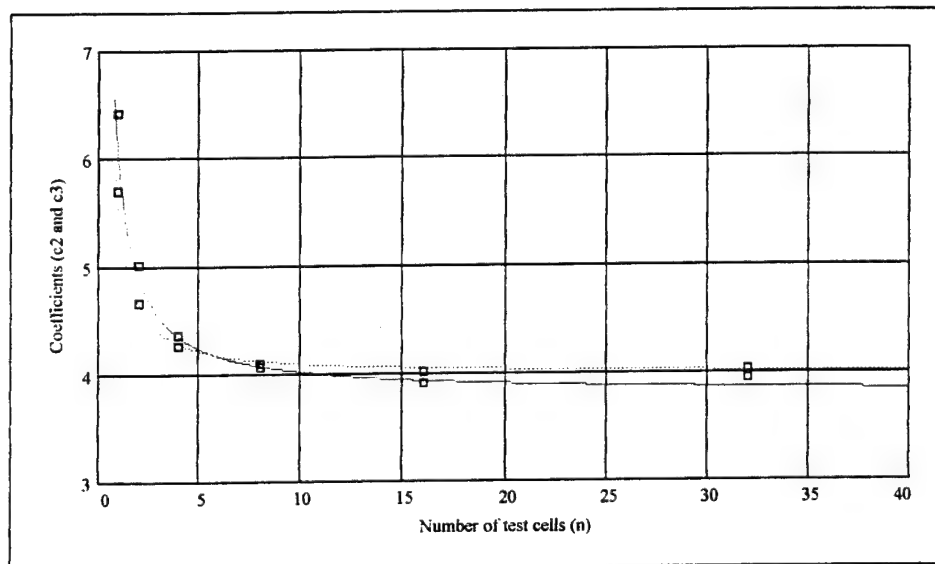


Figure 25. Coefficients C2 (Solid Line) and C3 (Dashed Line) as a Function of n
For Envelope Approximation: $a=1$, $b=1/2$

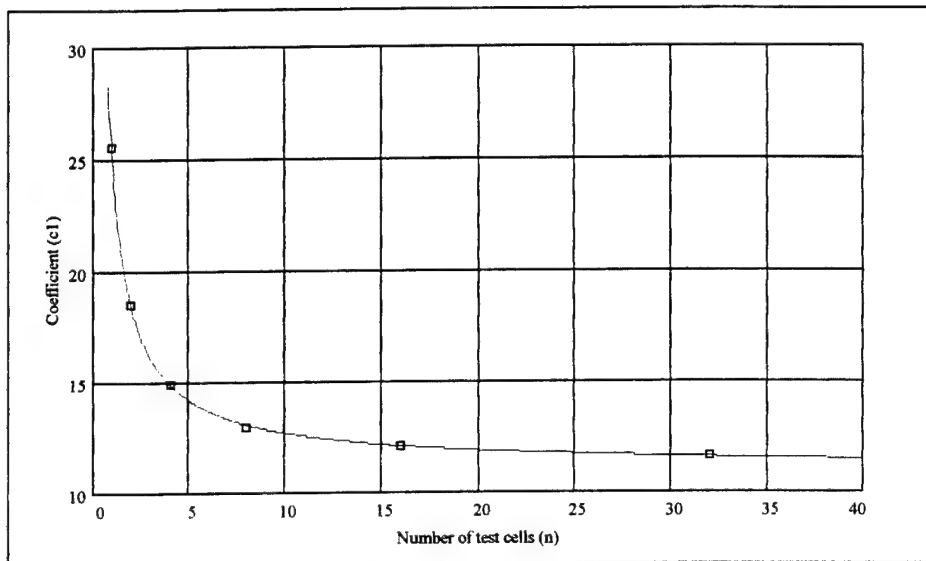


Figure 26: Coefficient C1 as a Function of n
For Envelope Approximation $a=1$, $b=1/4$

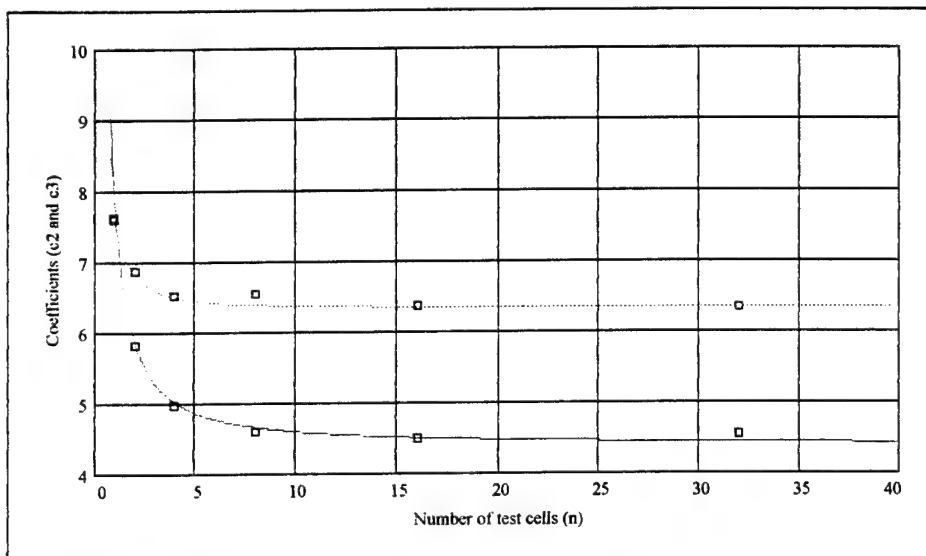


Figure 27. Coefficients C2 (Solid Line) and C3 (Dashed Line) as a Function of n
For Envelope Approximation: $a=1$, $b=1/4$

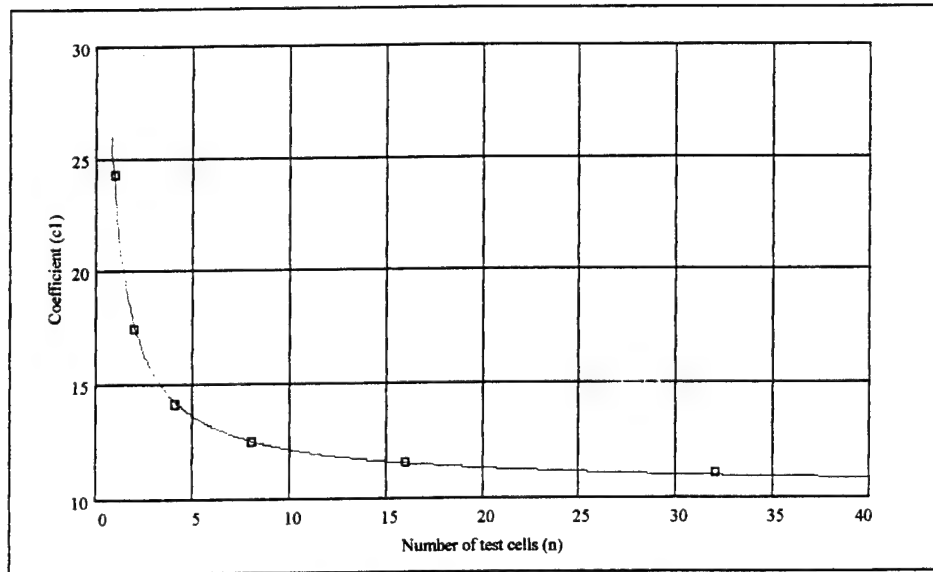


Figure 28: Coefficient C1 as a Function of n
For Envelope Approximation $a=1$, $b=3/8$

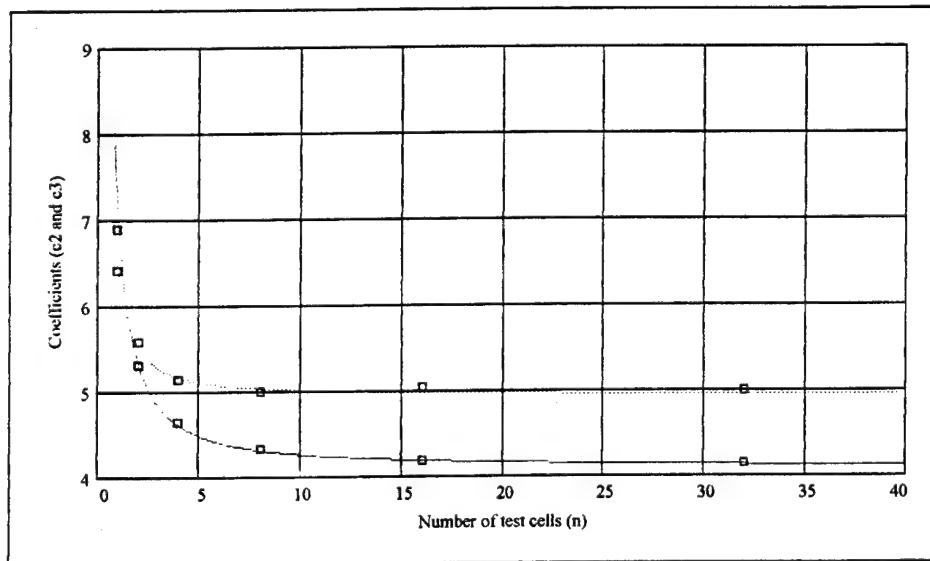


Figure 29. Coefficients C2 (Solid Line) and C3 (Dashed Line) as a Function of n
For Envelope Approximation: $a=1$, $b=3/8$

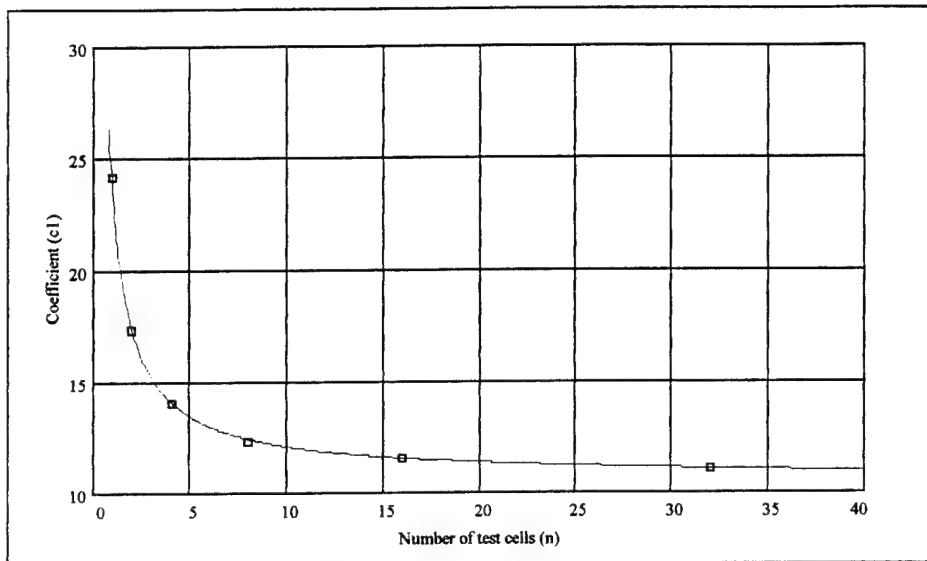


Figure 30: Coefficient C1 as a Function of n
For Envelope Approximation $a=31/32$, $b=3/8$

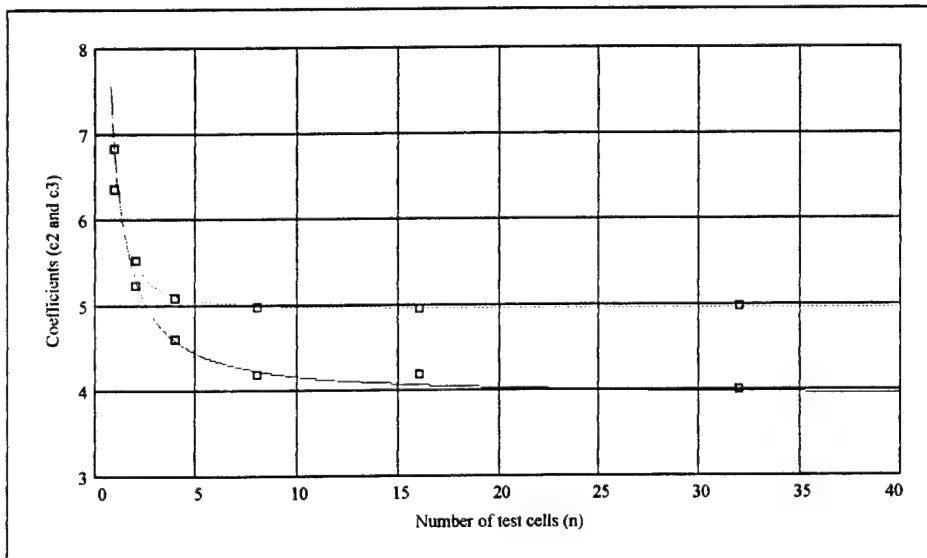


Figure 31. Coefficients C2 (Solid Line) and C3 (Dashed Line) as a Function of n
For Envelope Approximation: $a=31/32$, $b=3/8$

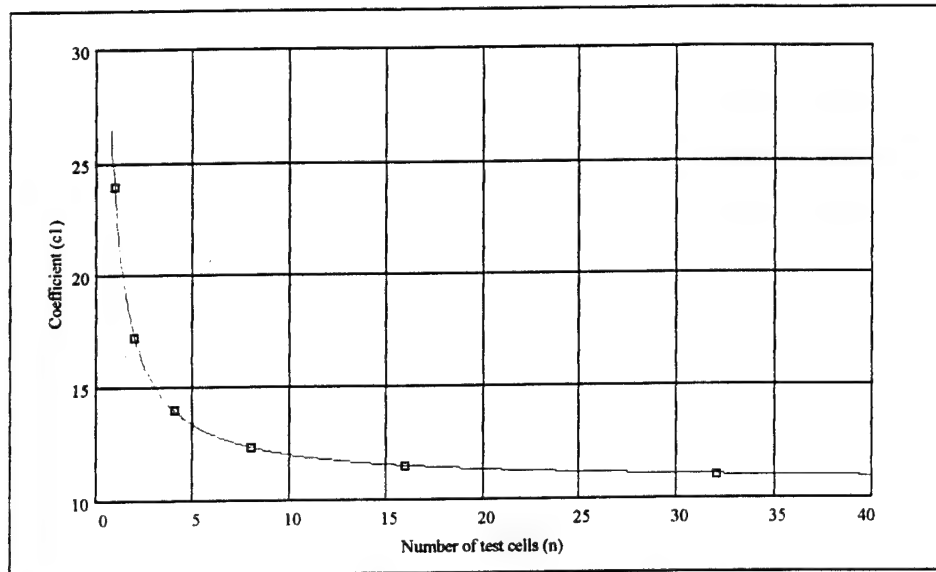


Figure 32: Coefficient C1 as a Function of n
For Envelope Approximation $a=0.948$, $b=393$

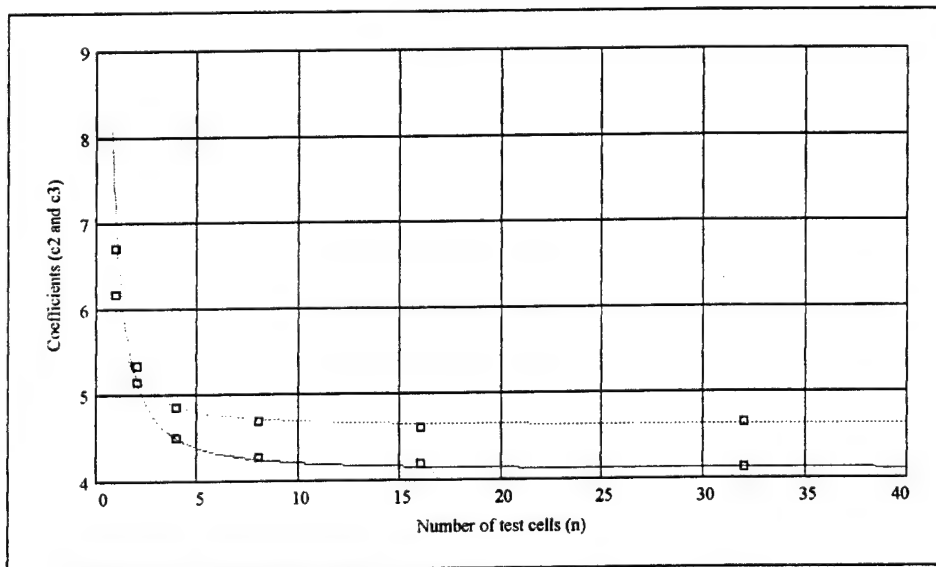


Figure 33: Coefficients C2 (Solid Line) and C3 (Dashed Line) as a Function of n
For Envelope Approximation: $a=0.948$, $b=0.393$

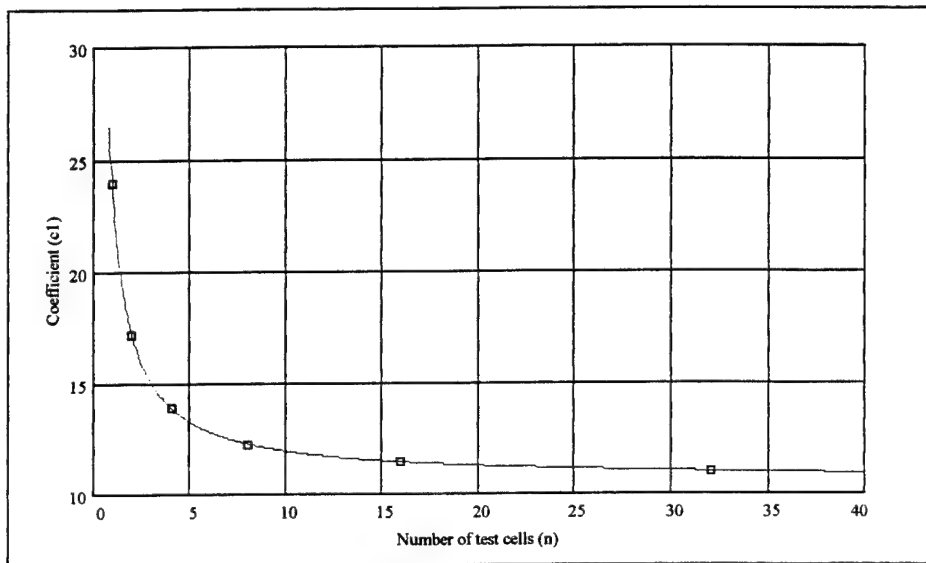


Figure 34: Coefficient C_1 as a Function of n
For Envelope Approximation $a=0.96043$, $b=0.39782$

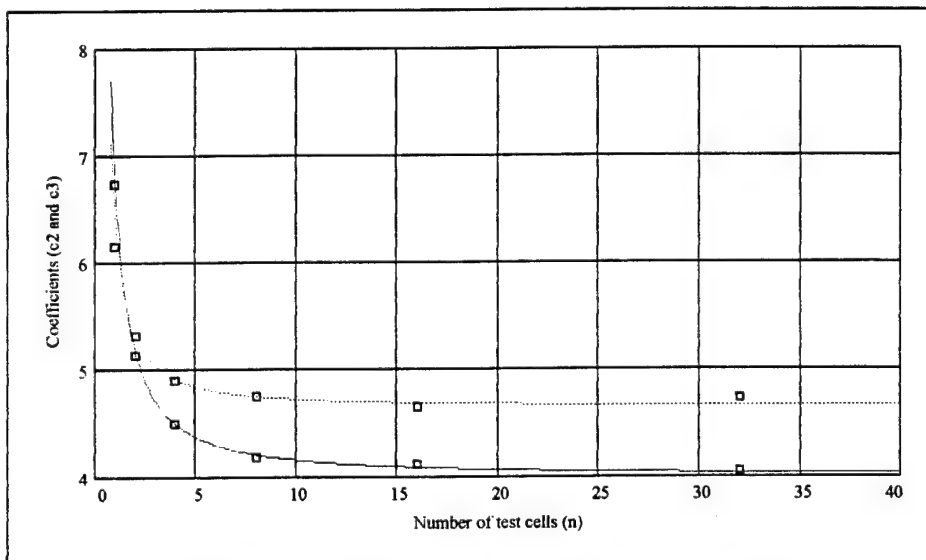


Figure 35. Coefficients C_2 (Solid Line) and C_3 (Dashed Line) as a Function of n
For Envelope Approximation: $a=0.96043$, $b=0.39782$

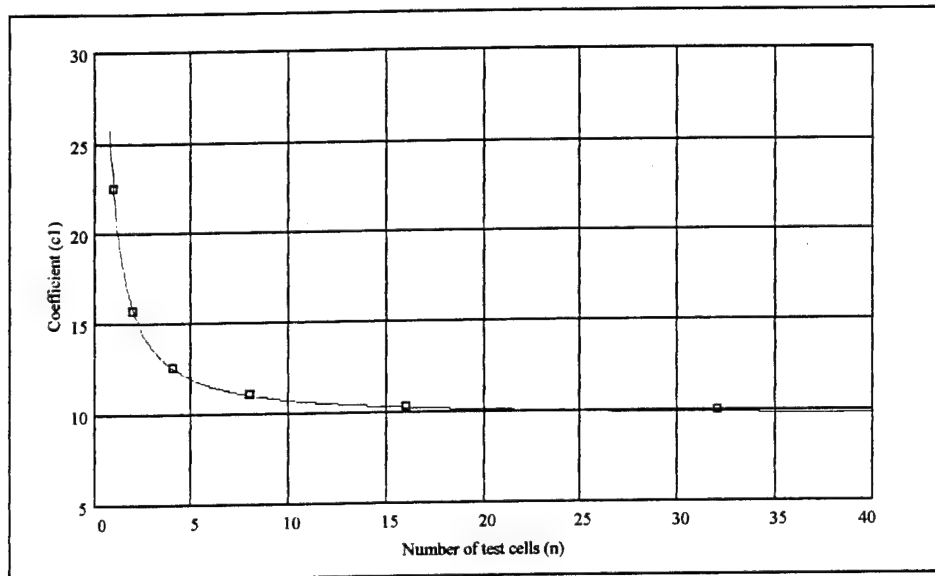


Figure 36: Coefficient C1 as a Function of n
For Envelope Detector

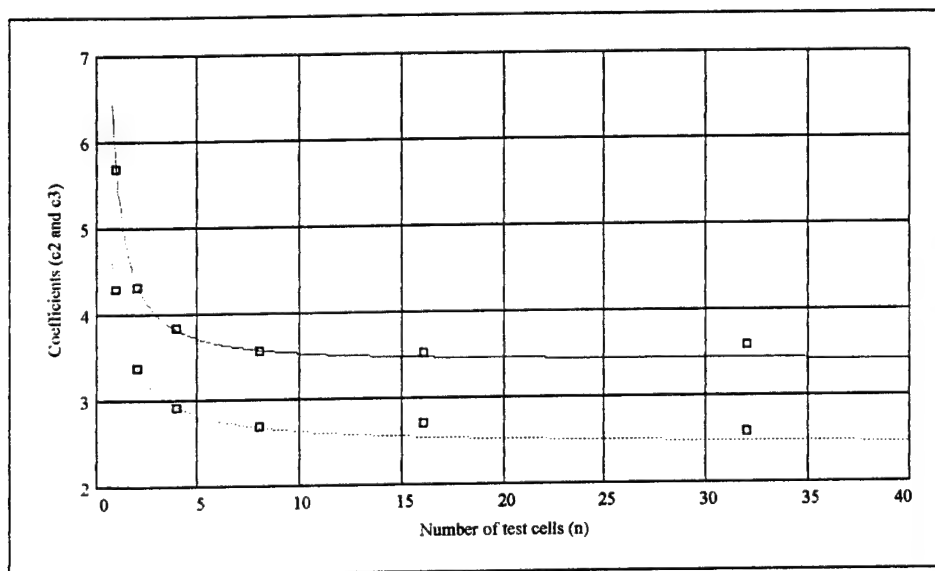


Figure 37. Coefficients C2 (Solid Line) and C3 (Dashed Line) as a Function of n
For Envelope Detector

VI. CONCLUSIONS

Monte Carlo simulations yield accurate PD vs. SNR curves for envelope detection approximation GO CFAR processors. These curves can be closely approximated with closed form expressions using the *erf* function in the form:

$$\text{PD} < 0.5 \quad PD = 0.5(\text{erf}(\frac{SNR-c1}{c2}) + 1) \quad (6.1)$$

$$\text{PD} > 0.5 \quad PD = 0.5(\text{erf}(\frac{SNR-c1}{c3}) + 1) \quad (6.2)$$

where the coefficients $c1$, $c2$ and $c3$ are chosen for a specific envelope approximation and number of reference cells n . These PD approximations compare favorably with the results of Monte Carlo simulations; the maximum residual in all cases is less than 0.028.

A curve fit of the coefficients $c1$, $c2$ and $c3$ as a function of the number of reference cells provides a means of quickly extracting the coefficients for any number of reference cells $1 < n \leq 32$ for a $PFA=10^{-4}$. In conjunction with the equations above, these coefficients produce accurate PD curves as a function of SNR.

APPENDIX - MATLAB PROGRAMS

Programs to Run Monte Carlo Simulation

runmont.m Batch file to run multiple simulations
setvar.m Program to set variable for each loop
montesim.m Main program to run simulations of envelope approximations
monteenv.m Main program to run simulation of envelope detector

Programs to run Plots of Data

plotmont.m Plots data from *.dat files
getdata.m Gets data to plot graphs

RUNMONT.M

```
% FILE NAME: RUNMONT.M

% Batch file to run all Monte Carlo Simulations for PFA=1e-4

clear
% Input parameters
%   a: coefficient a
%   b: coefficient b
%   n: number of cells per side
%   PFA: Required PFA
%   T: Threshold
%   N: number of monte carlo simulations per inner loop
%   MAXSNR: Maximum SNR in dB

%-----
% Set variables using data from pfa_a*.dat at PFA=1e-4
% SNR/PD data saved to file MC1_4.dat
set_var; % Clear all variables and reset to clear memory
a=1; b=1;
nTABLE=[ 1.000 2.000 4.000 8.000 16.00 32.000];
TTABLE=[12.287 5.952 4.423 3.891 3.680 3.5975];
montesim;
save MC1_4.dat mctemp -ascii
%-----
% Set variables using data from pfa_b*.dat at PFA=1e-4
% SNR/PD data saved to file MC2_4.dat
set_var; % Clear all variables and reset to clear memory
a=1; b=0.5;
nTABLE=[ 1.000 2.000 4.000 8.0000 16.00 32.000];
TTABLE=[13.430 6.515 4.773 4.1525 3.903 3.8040];
montesim;
save MC2_4.dat mctemp -ascii
%-----
% Set variables using data from pfa_c*.dat at PFA=1e-4
% SNR/PD data saved to file MC3_4.dat
set_var; % Clear all variables and reset to clear memory
a=1; b=0.25;
nTABLE=[ 1.000 2.000 4.000 8.0000 16.00 32.000];
TTABLE=[16.685 7.915 5.530 4.6750 4.335 4.197];
montesim;
save MC3_4.dat mctemp -ascii
%-----
% Set variables using data from pfa_d*.dat at PFA=1e-4
% SNR/PD data saved to file MC4_4.dat
set_var; % Clear all variables and reset to clear memory
a=1; b=3/8;
nTABLE=[ 1.000 2.000 4.000 8.0000 16.00 32.000];
TTABLE=[14.530 7.018 5.064 4.3610 4.076 3.9636];
```



```

montesim;
save MC4_4.dat mctemp -ascii
%-----
% Set variables using data from pfa_e*.dat at PFA=1e-4
% SNR/PD data saved to file MC5_4.dat
set_var; % Clear all variables and reset to clear memory
a=31/32; b=3/8;
nTABLE=[ 1.000 2.000 4.000 8.0000 16.00 32.000];
TTABLE=[14.399 6.960 5.032 4.3380 4.058 3.9465];
montesim;
save MC5_4.dat mctemp -ascii
%-----
% Set variables using data from pfa_f*.dat at PFA=1e-4
% SNR/PD data saved to file MC6_4.dat
set_var; % Clear all variables and reset to clear memory
a=0.948; b=0.393;
nTABLE=[ 1.000 2.000 4.000 8.0000 16.00 32.00];
TTABLE=[14.112 6.835 4.960 4.2870 4.017 3.907];
montesim;
save MC6_4.dat mctemp -ascii
%-----
% Set variables using data from pfa_g*.dat at PFA=1e-4
% SNR/PD data saved to file MC7_4.dat
set_var; % Clear all variables and reset to clear memory
a=0.96043; b=0.39782;
nTABLE=[ 1.000 2.000 4.000 8.0000 16.00 32.00];
TTABLE=[14.114 6.834 4.960 4.2880 4.017 3.908];
montesim;
save MC7_4.dat mctemp -ascii
%-----
% Set variables using data from pfa_r*.dat at PFA=1e-4
% SNR/PD data saved to file MCr_4.dat
set_var; % Clear all variables and reset to clear memory
nTABLE=[ 1.000 2.000 4.000 8.0000 16.000 32.000];
TTABLE=[11.962 5.751 4.251 3.7290 3.5215 3.4395];
monteenv;
save MCr_4.dat mctemp -ascii

```

SET_VAR.M

% FILE NAME: SET_VAR.M

% File to set variables

clear global

clear all

% Input parameters

% a: coefficient a

% b: coefficient b

% n: number of cells per side

% PFA: Required PFA

% T: Threshold

% Nmax: Max number of monte carlo simulations per angle phi

% MAXSNR: Maximum SNR in dB

angle=pi/2;% Set max angle to sample (0 to pi/2)

phi=0:angle/40:angle;% Set phi to vector of angles from 0 to pi/2, 40 steps

PFA=1e-4;

Nmax=10000;

MAXSNR=40;

MONTESIM.M

```
% FILE NAME: MONTESIM.M
% Generic Monte Carlo Simulation
%
% Input parameters
%   a: coefficient a
%   b: coefficient b
%   n: number of cells per side
%   PFA: Required PFA
%   T: Threshold
%   Nmax: Max number of monte carlo simulations per angle phi
%   N: Size of random matrix (nxN)=Nmax
%   MAXSNR: Maximum SNR in dB
%   nTABLE: Values of n=number of cells per side
%   TTABLE: Values of T=corresponding threshold setting for given n and PFA
%   angle: Max angle to sample (0 to pi/2)
%   phi: vector of angles from 0 to pi/2, 40 steps

for num=1:length(nTABLE);
    skip=0; % Sets flag to zero (Used to remove unnecessary loops)
    count=0; % Sets count to zero (Used to remove unnecessary loops)
    n=nTABLE(num);
    T=TTABLE(num);
    N=round(Nmax/n); % Size of random matrix (nxN)=Nmax

    SNRdB=0:0.5:MAXSNR;% Set Signal-to-Noise ratio in dBs (increments of 0.5 dB)
    PD=zeros(1,length(SNRdB));

    %-----
    for i=1:length(SNRdB); % Performs loop over all SNRs
        if skip==1
            break
        end

        for j=1:length(phi), % Performs loop over all angles phi: 0 to pi/4 (20 steps)
            SNR=10^(SNRdB(i)/10);% Converts SNR from dBs to linear scale [1,1]
            A=sqrt(2*SNR); % Converts SNR value to required A [1,1]
            In1=randn(n,N); % I on left side, noise only [nxN]
            In2=randn(n,N); % I on right side,noise only [nxN]
            Qn1=randn(n,N); % Q on left side, noise only [nxN]
            Qn2=randn(n,N); % Q on left side, noise only [nxN]
            if n==1
                y1=a*abs(In1)+b*abs(Qn1);
                y2=a*abs(In2)+b*abs(Qn2);
            else
                y1=a*sum(abs(In1))+b*sum(abs(Qn1)); % Sum of left side noise inputs [1xN]
                y2=a*sum(abs(In2))+b*sum(abs(Qn2)); % Sum of right side noise inputs [1xN]
            end
        end
    end
end
```

```

z=max(y1,y2);% GO output [1xN]
Vt=T*z/n;      % Comparison voltage [1xN]
%-----
% Test voltage in test cell with signal [1,N]
% z=a*|I|+b*|Q| or z=a*|Acos(phi)+x|+b*|Asin(phi)+y|
Vtest=a*abs(A*cos(phi(j))+randn(1,N))+b*abs(A*sin(phi(j))+randn(1,N));
%-----
% Calculate percentage
% Add up all percentages for each value of phi
PD(i)=PD(i)+length(find((Vtest-Vt)>0))/(N*length(phi)); % [1 x # of SNRs]
end
% deletes unnecessary loops if PD=1 for greater SNRs
if PD(i)>=1
    count=count+1;
    if count>5
        skip=1;
    end
end
end
end

%----- Break branches to here

for k=1:length(SNRdB),
    if k>=i
        PD(k)=1;
    end
end
% Store SNR/PD data to file named mctemp (temporarily)
mctemp((2*num)-1:2*num,:)= [SNRdB;PD];
end

```

MONTEENV.M

```
% FILE NAME: MONTEENV.M
% Generic Monte Carlo Simulation for envelope detector
%
% Input parameters
%   n: number of cells per side
%   PFA: Required PFA
%   T: Threshold
%   Nmax: Max number of monte carlo simulations per angle phi
%   N: Size of random noise matrix (nxN)=Nmax
%   MAXSNR: Maximum SNR in dB
%   nTABLE: Values of n=number of cells per side
%   TTABLE: Values of T=corresponding threshold setting for given n and PFA
%   angle: Max angle to sample (0 to pi/2)
%   phi: vector of angles from 0 to pi/2, 40 steps

for num=1:length(nTABLE);
    skip=0; % Sets flag to zero
    count=0;% Sets count to zero
    n=nTABLE(num);
    T=TTABLE(num);
    N=round(Nmax/n); % Set size of random noise matrix (nxN)=Nmax

    SNRdB=0:0.5:MAXSNR;% Set Signal-to-Noise ratio in dBs (increments of 0.5 dB)
    PD=zeros(1,length(SNRdB));
    %-----
    for i=1:length(SNRdB), % Performs loop over all SNRs
        if skip==1
            break
        end

        for j=1:length(phi), % Performs loop over all angles phi: 0 to pi/4 (20 steps)
            SNR=10^(SNRdB(i)/10); % Converts SNR from dBs to linear scale [1,1]
            A=sqrt(2*SNR); % Converts SNR value to required A [1,1]
            In1=randn(n,N); % I on left side, noise only [nxN]
            In2=randn(n,N); % I on right side,noise only [nxN]
            Qn1=randn(n,N); % Q on left side, noise only [nxN]
            Qn2=randn(n,N); % Q on left side, noise only [nxN]
            if n==1
                y1=sqrt((In1.^2)+(Qn1.^2));
                y2=sqrt((In2.^2)+(Qn2.^2));
            else
                y1=sum(sqrt((In1.^2)+(Qn1.^2))); % Sum of left side noise inputs [1xN]
                y2=sum(sqrt((In2.^2)+(Qn2.^2))); % Sum of right side noise inputs [1xN]
            end
            z=max(y1,y2); % GO output [1xN]
            Vt=T*z/n; % Comparison voltage [1xN]
            %-----
            % Test voltage in test cell with signal [1,N]
        end
    end
end
```

```

% z=sqrt[(I^2)+(Q^2) or z=sqrt[(Acos(phi)+x)^2+(Asin(phi)+y)^2]
Vtest=sqrt((A*cos(phi(j))+randn(1,N)).^2+(A*sin(phi(j))+randn(1,N)).^2);
%-----
% Calculate percentage
% Add up all percentages for each value of phi
PD(i)=PD(i)+length(find((Vtest-Vt)>0))/(N*length(phi)); % [1 x # of SNRs]
end
% deletes unnecessary loops if PD=1 for greater SNRs
if PD(i)>=1
    count=count+1;
    if count>5
        skip=1;
    end
end
end
end

%---- Break branches to here

for k=1:length(SNRdB),
    if k>=i
        PD(k)=1;
    end
end
% Store SNR/PD data to file named mctemp (temporarily)
mctemp((2*num)-1:2*num,:)= [SNRdB;PD];
end

```

PLTMONT.M

```
% FILE NAME: PLTMONT.M
% Batch file to plot PD vs. SNR curves from Monte Carlo Simulations
% Plot PD vs SNR curve from mc1_4.dat
load mc1_4.dat
save mc.dat mc1_4 -ascii
var_title='PD vs. SNR a=1,b=1,PFA=1e-4, n=1,2,4,8,16,32';
get_data
% Plot PD vs SNR curve from mc2_4.dat
load mc2_4.dat
save mc.dat mc2_4 -ascii
var_title='PD vs. SNR a=1,b=1/2,PFA=1e-4, n=1,2,4,8,16,32';
get_data
% Plot PD vs SNR curve from mc3_4.dat
load mc3_4.dat
save mc.dat mc3_4 -ascii
var_title='PD vs. SNR a=1,b=1/4,PFA=1e-4, n=1,2,4,8,16,32';
get_data
% Plot PD vs SNR curve from mc4_4.dat
load mc4_4.dat
save mc.dat mc4_4 -ascii
var_title='PD vs. SNR a=1,b=3/8,PFA=1e-4, n=1,2,4,8,16,32';
get_data
% Plot PD vs SNR curve from mc5_4.dat
load mc5_4.dat
save mc.dat mc5_4 -ascii
var_title='PD vs. SNR a=31/32,b=3/8,PFA=1e-4, n=1,2,4,8,16,32';
get_data
% Plot PD vs SNR curve from mc6_4.dat
load mc6_4.dat
save mc.dat mc6_4 -ascii
var_title='PD vs. SNR a=0.948,b=0.393,PFA=1e-4, n=1,2,4,8,16,32';
get_data
% Plot PD vs SNR curve from mc7_4.dat
load mc7_4.dat
save mc.dat mc7_4 -ascii
var_title='PD vs. SNR a=0.96043,b=0.39782,PFA=1e-4, n=1,2,4,8,16,32';
get_data
% Plot PD vs SNR curve from mcr_4.dat
load mcr_4.dat
save mc.dat mcr_4 -ascii
var_title='PD vs. SNR for envelope detector,PFA=1e-4, n=1,2,4,8,16,32';
get_data
```


GET_DATA.M

```
% FILE NAME: GET_DATA.M
% Batch file to plot PD vs. SNR curves from Monte Carlo Simulations

% Get data
clf
load mc.dat
num_plot=length(mc(:,1))/2;
hold on
grid on
for i=1:num_plot;
    plot(mc((2*i)-1,:),mc((2*i),:))
end
xlabel('Signal-to-Noise Ratio (dB)')
ylabel('Probability of Detection')
title(var_title)
% print -dwin
pause
```


REFERENCES

- Daniel, C., and Wood, F., *Fitting Equations to Data*, John Wiley & Sons, 1980.
- Filip, A. E., "A Baker's Dozen Magnitude Approximations and Their Detection Statistics." *IEEE Transactions on Aerospace and Electronic Systems*, AES-12 (Jan. 1976), 86-89.
- Hache, J.J.P., *Probability of Detection for a GO CFAR Radar Processor Using Envelope Detection Approximation*, Naval Postgraduate School Master's Thesis (Sept. 1994).
- Mathsoft, Inc., *Mathcad User's Guide Windows Version*, 1992.
- Pace, P., and Taylor, L., "False Alarm Analysis of the Envelope Detection GO-CFAR Processor." *IEEE Transactions on Aerospace and Electronic Systems*, Vol. 30, No. 3, (July, 1994).
- Papoulis, A., *Probability, Random Variables, and Stochastic Processes*, McGraw-Hill, 1984.
- Rubinstein, R., *Simulation and the Monte Carlo Method*, John Wiley & Sons, 1981.
- Skolnik, M. I., *Introduction to Radar Systems*, McGraw-Hill, Inc., 1980.
- The MathWorks, Inc., *MATLAB User's Guide*, 1992.
- Wilson, D.J., "Probability of Detection of the Sum of the Magnitudes of In-Phase and Quadrature Signals in Gaussian Noise." *General Dynamics Technical Memorandum*, TM:6-257-640, (Apr. 1982).

INITIAL DISTRIBUTION LIST

1. Defense Technical Information Center
Cameron Station
Alexandria, Virginia 22304-6145 2
2. Library, Code 52
Naval Postgraduate School
Monterey, California 93943-5101 2
3. Chairman, Code AA
Department of Aeronautics and Astronautics
Naval Postgraduate School
Monterey, California 93943-5000 1
4. Chairman, Code ECE
Department of Electrical and Computer Engineering
Naval Postgraduate School
Monterey, California 93943-5121 1
5. P.E. Pace, Code EC/PC
Department of Electrical and Computer Engineering
Naval Postgraduate School
Monterey, California 93943-5121 2
6. E. M. Wu, Code AA/WU
Department of Aeronautics and Astronautics
Naval Postgraduate School
Monterey, California 93943-5000 2
7. Hughes Missile Systems Company, Tucson
Research Engineering
Attn: Mr. L Lamoyne Taylor
Bldg. 842 MS-3
P.O. Box 11337
Tucson, Arizona 85734 1
8. Hughes Missile Systems Company, Tucson
Phalanx Program Office
Attn: Mr. Lorne B. Smith
Bldg. 842
P.O. Box 11337
Tucson, Arizona 85734 1

9. Hughes Missile Systems Company, Tucson
Phalanx Program Office
Attn: Mr. Peter M. McCray
Bldg. 842
P.O. Box 11337
Tucson, Arizona 85734 1
7. LT C. Tanaka
Patrol Squadron Special Projects Unit Two
NAS Barbers Pointt, Hawaii 96862-6100 2